NASA Contractor Report 172376

NASA-CR-172376
19850001700

Development and Application of Algorithms for Calculating the Transonic Flow About Harmonically Oscillating Wings

F. Edward Ehlers, Warren H. Weatherill, and Elizabeth L. Yip

The Boeing Commercial Airplane Company
Seattle, Washington

Contract NAS1-16297
October 1984

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665
AC 804 827-3966

NASA Contractor Report 172376

Development and Application of Algorithms for Calculating the Transonic Flow About Harmonically Oscillating Wings

F. Edward Ehlers, Warren H. Weatherill, and Elizabeth L. Yip

The Boeing Commercial Airplane Company
Seattle, Washington

CONTENTS

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Concluded)

1.0 SUMMARY

A finite difference method for solving the unsteady transonic flow about harmonically oscillating wings is investigated. The procedure is based on separating the velocity potential into steady and unsteady parts and linearizing the resulting unsteady differential equation for small disturbances. The differential equation for the unsteady velocity potential is linear with spatially varying coefficients and with the time variable eliminated by assuming harmonic motion.

The work of this report is a direct extension of earlier studies and includes correlation with experimental results for two rectangular wings and investigations of possible solution techniques for three-dimensional wings of more general planform.

The main results of the study are as follows:

1. An alternating direction implicit (ADI) procedure is investigated and a pilot program is developed for both two- and three-dimensional wings. This program provides a relatively efficient relaxation solution without previously encountered solution instability problems.

2. Pressure distributions for two rectangular wings are calculated and the results correlated with experimental data.

3. Conjugate gradient techniques are developed for the asymmetric, indefinite problem. The conjugate gradient procedure is evaluated for applications to the unsteady transonic problem. Several preconditioning methods are investigated.

4. Difference equations for the alternating direction procedure are derived using a coordinate transformation for swept and tapered wing planforms. For a coordinate transformation which is continuous up to the second derivative, the ADI method converged for sweep angles up to 45 deg. The pressure distribution for the swept, untapered flat plate is correlated with the kernel function method.

The results for the conjugate gradient method are preliminary and further testing of the method is contemplated. Of the techniques studied, the most efficient procedure for the analysis of three-dimensional wings on computers of limited memory (i.e., on the order of one million words or less) appears to be the ADI method.

2.0 INTRODUCTION

The development of a capability for calculating unsteady transonic airforces for use in flutter analysis continues to be of interest. The considerable effort on this subject falls generally into two categories; (1) finite difference solutions of the transonic small perturbation equation for the velocity potential and (2) a modified strip theory procedure in which the subsonic coefficients are corrected empirically for transonic flow characteristics. In the first category, there are two main approaches; the first consists of a time integration of the nonlinear differential equations for the unsteady velocity potential, and the second involves separating the velocity potential into steady and harmonically varying unsteady parts. The latter is the approach we have used, and its formulation results in the usual nonlinear differential equation for the steady velocity potential and a linear equation for the unsteady velocity potential, with spatially varying coefficients which are functions of the steady velocity potential. Developments following this last approach have been documented in a series of NASA CRs (refs. 1 through 7). Results have shown that the two-dimensional, typical section problem can be handled by this harmonic approach relatively efficiently and accurately. However, the direct numerical solution technique which is successful for two-dimensions appears to be expensive when used on the full three-dimensional problem.

The purpose of the work discussed in this report is to explore the feasibility of solving the three-dimensional problem. This was accomplished by investigating several different procedures. The first is a finite difference formulation which results in a set of equations to be solved by ADI relaxation techniques. Then conjugate gradient techniques were investigated. Finally, the out-of-core direct solution module was rewritten from two dimensions to three dimensions and transferred from the CYBER to the CRAY.

The ADI method was derived by expressing the time-dependent differential equation for small perturbation transonic flow in difference equations using the alternating direction implicit (ADI) technique and then making the assumption of harmonic motion. The resulting ADI procedure does not have the frequency limitation on convergence, as do the standard block relaxation techniques. The ADI method was first tried on the two-dimensional problem. Correlations of examples for airfoils of vanishing thickness and for finite thickness airfoils are presented in section 5 with some discussion of convergence rates. The method was also extended to the rectangular wing for which the equations are derived in appendix A. Results for configurations of vanishing thickness are correlated with results from the integral equation method which is described in section 6 of reference 19. Also, results for rectangular wings with a 5%-thick circular arc airfoil and with a 12%-thick supercritical airfoil are correlated with experimental measurements. Problems in convergence were encountered when the method was applied to swept wings using a coordinate transformation aligning the streamwise variable with the leading and trailing edges of the wing. This was alleviated by using a swept wing coordinate transformation having continuous second derivatives across boundaries of mapping regions. Solutions for the swept untapered wing were obtained for sweep angles up to 45 deg. and correlated with the integral equation method of references 8 and 9. The solution for a 50 deg. swept untapered wing failed to converge.

The classical conjugate gradient method is for solving systems of linear equations of the form $Ax = b$ with coefficient matrices which are symmetric and positive definite. This procedure proved to be efficient for large, sparse, well conditioned systems. Work for this report has resulted in an algorithm for system matrices which need only to be nonsingular. Before applying the conjugate gradient algorithm, the coefficient matrix for the transonic problem must be preconditioned in order to speed convergence. Three methods of preconditioning were tried and are discussed in section 7. The most effective procedure is derived from the partial LU decomposition of the coefficient matrix. The derivation of the complete algorithm is presented in appendix E.

The pilot program for the out-of-core direct solution of two-dimensional airfoils was extended to rectangular three-dimensional wings. This pilot program was also converted from the CYBER to the CRAY computer to take advantage of the significantly larger core storage. The resulting test runs were expensive, but the cost was dominated by charges for input/output operations. The availability of machines with even larger core storage capability, and the characteristic that a large number of mode shapes may be handled for a minimal additional cost over that for one mode, warrant further work on the direct solution.

3.0 ABBREVIATIONS AND SYMBOLS

a	Amplitude of wing oscillations
b	Root semichord
C_P	Pressure coefficient, $(p-p_0)/(\frac{1}{2}\rho_0 U_0^2)$ where p is the local pressure, p_0 the freestream static pressure, and ρ_0 the freestream air density
A	Matrix of coefficients, also aspect ratio (Section 6)
f	Frequency in Hertz
f_0	Undisturbed wing or airfoil shape
f_1	Unsteady contribution to wing or airfoil shape
i,j I,J	x,y subscripts and indices for points in the mesh
i	$\sqrt{-1}$
I	Identity matrix
k	Reduced frequency based on semichord, 2π fb/U; same as ω.
K	Transonic parameter, $(1-M^2)/(M^2\varepsilon)$
le	Leading edge
M	Freestream Mach number
n	(n_x, n_y, n_t), unit normal vector to shock
q	$\omega^2/\varepsilon - i\omega(\gamma - 1)\varphi_{0_{xx}}$
t	Time in units of b/U
te	Trailing edge
u	$K - (\gamma + 1)\varphi_{0_x}$
U,U_0,V	Freestream velocity
x	Freestream coordinate. Vector of unknowns in matrix equations.
x_0,y_0	Physical coordinates, made dimensionless with the root semichord

x,y	Scaled coordinates $(x_0,\mu y_0)$ for the two-dimensional problem
X_0	Steady chordwise shock location
X_1	Complex amplitude of shock oscillation
X_m	Magnitude of X_1
α	Angle of attack
β_1	$= e^{-i\omega\Delta t}$
β_2	$= <u_x>$
γ	Ratio of specific heats for air
ΔC_p	Jump in pressure coefficient across airfoil or wake
Δt	Time step in ADI procedure
$\Delta\varphi_1$	Jump in φ_1 at plane of wing or vortex wake
$\Delta\varphi_{1_{te}}$	Jump in φ_1 at wing trailing edge
δ	Thickness ratio; also finite difference operator
ε	$(\delta/M)^{2/3}$
λ_1	$\omega M/(1-M^2)$
μ	Scale factor of y_0, $\mu = \delta^{1/3} M^{2/3}$
$\bar{\eta}$	Fraction of semichord
ξ,η,ζ	Swept wing coordinates
ϕ	Unsteady time dependent perturbation potential
φ, φ	Complete, scaled perturbation velocity potential; also used for the unsteady potential in finite difference equations with subscripts
φ_0, φ_0	Steady scaled perturbation velocity potential
φ_1, φ_1	Unsteady scaled perturbation velocity potential
ω	Reduced frequency in radians; same as k

Notation

[ ]	Denotes jump in quantity across shock
< >	Denotes mean value of quantity at shock
Δ	Denotes jump in quantity across airfoil except for Δt

4.0 FORMULATION AND SOLUTION

Since the mathematical derivation of the method for the solution of the unsteady velocity potential for the flow about a harmonically oscillating wing is presented in reference 1, the discussion here will be limited to a brief outline of the procedure for two dimensions. The complete nonlinear differential equation was simplified by assuming the flow to be a small perturbation from a uniform stream near the speed of sound. The resulting equation for unsteady flow is

$$\left[K - (\gamma - 1)\,\varphi_t - (\gamma + 1)\,\varphi_x\right]\varphi_{xx} + \varphi_{yy} - \left(2\varphi_{xt} + \varphi_{tt}\right)/\epsilon = 0 \tag{4.1}$$

where $K = (1\text{-}M^2)/(M^2\varepsilon)$, M is the freestream Mach number of velocity U_o in the x-direction, x and y are made dimensionless to the semichord b of the airfoil and the time t to the ratio b/U_o. With the airfoil shape as a function of time defined by the relation

$$y_0 = \delta f(x,t)$$

the linearized boundary condition becomes

$$\varphi_y = f_x\,(x,t) + f_t\,(x,t) \tag{4.2}$$

The quantity δ is associated with properties of the airfoil (such as maximum thickness ratio, camber, or maximum angle of attack) and is assumed to be small. The coordinate y is scaled to the dimensionless physical coordinate y_o according to

$$y = \delta^{1/3}\,M^{2/3}y_0$$

and ε is given in terms of δ by

$$\epsilon = (\delta/M)^{2/3}$$

The pressure coefficient is found from the relation

$$C_p = -\,2\epsilon\,(\varphi_x + \varphi_t)$$

The preceding differential equation is simplified by assuming harmonic motion and by assuming the velocity potential to be separable into a steady-state potential and a potential representing the unsteady effects. We write for a perturbation velocity potential

$$\varphi = \varphi_0\,(x,y) + \varphi_1\,(x,y)\,e^{i\omega t} \tag{4.3}$$

and for the body shape

$$y_0 = \delta f\,(x,t) = \delta\left[f_0\,(x) + f_1\,(x)\,e^{i\omega t}\right]$$

Since the steady-state terms must satisfy the boundary conditions and the differential equation in the absence of oscillations, we obtain

$$\left[K - (\gamma + 1)\,\varphi_{0_x}\right]\varphi_{0_{xx}} + \varphi_{0_{yy}} = 0 \tag{4.4}$$

with

$$\varphi_{0_y} = f_0\,(x), \qquad y = 0, \qquad -1 \le x \le 1 \tag{4.5}$$

On the assumption that the oscillations are small and products of φ_1 may be neglected, equations (4.1) and (4.2) with the aid of equations (4.4) and (4.5) yield

$$\{[K - (\gamma + 1)\,\varphi_{0_x}]\,\varphi_{1_x}\}_x + \varphi_{1_{yy}} - (2i\omega/\epsilon)\,\varphi_{1_x} + q\varphi_1 = 0 \tag{4.6}$$

where

$$q = \omega^2/\epsilon - i\omega\,(\gamma - 1)\,\varphi_{0_{xx}}$$

subject to the wing boundary conditions

$$\varphi_{1_y} = f_{1_x} + i\omega f_1\,(x), \qquad y = 0, \qquad -1 \le x \le 1 \tag{4.7}$$

The boundary condition that the pressure be continuous across the wake from the trailing edge was found in terms of the jump in potential $\Delta\varphi_1$ to be

$$\Delta\varphi_1 = \Delta\varphi_{1_{te}} e^{-i\omega\,(x - x_{te})} \tag{4.8}$$

where $\Delta\varphi_{1_{te}}$ is the jump in the potential at $x = x_{te}$ just downstream of the trailing edge and is determined to satisfy the Kutta condition that the jump in pressure vanish at the trailing edge. The quantity $\Delta\varphi_1$ is also used in the difference formulation for the derivative $\varphi 1_{yy}$ to satisfy continuity of normal flow across the trailing edge wake.

For the set of difference equations to be determinate, the boundary conditions on the outer edges of the mesh must be specified. In the original unsteady formulation, these boundary conditions were derived from asymptotic integral relations in a manner parallel to that used by Klunker (ref. 10) for steady flow. A later formulation (ref. 3) applies an outgoing plane wave boundary condition to the outer edges of the mesh. This boundary condition is numerically simpler to apply and is equivalent to the first order nonreflecting boundary conditions derived by Engquist and Majda (ref. 11).

A computer program for solving the steady state transonic flow about lifting airfoils based on equations (4.4) and (4.6) was developed by Cole, Murman, and Krupp (refs. 12 and 13). Steady-state solutions required for the coefficients of equation (4.6) were obtained by using the latest version of this program, TSFOIL (reference 14), and also by using the steady-state solutions from the time dependent method of Rizzetta and Chin (ref. 15) called EXTRAN2. An additional improvement on locating the shock is also available both in TSFOIL and EXTRAN2. For airfoils at high Mach numbers and angles of attack, TSFOIL and EXTRAN2 predict shock positions considerably aft of experimental measurements. To overcome this difficulty, Jou and Murman (ref. 16) developed a phenomenological model for the displacement thickness effects of shock wave boundary layer interactions. A wedge (or ramp) is introduced behind the shock to simulate the thickening of the boundary layer. This procedure was not needed for the results reported here, however. For three dimensions, we used XTRAN3S, the time dependent method of Borland, Rizzetta, and Yoshihara in reference 17, to compute the required steady state potential.

The similarity of the unsteady differential equation to the steady state equation suggests that the method of Cole, Murman, and Krupp should be an effective way to solve equation (4.6) for the unsteady potential φ_1. Note that equation (4.6) is of mixed type, being elliptic or hyperbolic whenever equation (4.4) is elliptic or hyperbolic. Central differencing was used at all points for the y derivative and all subsonic or elliptic points for the x derivatives. Backward (or upstream) differences were used for the x derivatives at all hyperbolic points. The preferred

numerical approach to solving the resulting large-order set of difference equations is a relaxation procedure, which permits the calculation to be made as a sequence of relatively small problems. However, as discussed in preceding NASA reports by the authors (refs. 3 and 4), a significant problem of solution convergence with the relaxation procedure was encountered that severely limits the range of Mach number and reduced frequency for which solutions may be obtained. Accordingly, an out-of-core solver (ref. 18) was developed to solve the complete set of difference equations simultaneously, which for two-dimensional flow is relatively efficient.

The size of practical three-dimensional problems is such that the out-of-core direct solver would appear to be very expensive. This and other solution procedures are under investigation. First, conjugate gradient techniques have been examined as a more efficient means for obtaining a direct solution. Since the matrix of coefficients is neither symmetric, nor always positive definite, nor well conditioned, special procedures must be used in applying the conjugate gradient technique. Discussion of the algorithms tried during the current work is presented in section 7. Second, a relaxation-type method based on the time-dependent ADI method to obtain frequency domain solutions was derived which does not have the frequency limitations of classical block relaxation. If, in place of substituting equation (4.3) into equation (4.1), we use instead

$$\varphi(x,y,t) = \varphi_0 (x,y) + \varphi_1 (x,y,t)$$

we obtain the following linear differential equation for φ_1:

$$\{[K - (\gamma + 1)\, \varphi_{0_x}]\, \varphi_{0_x}\}_x + \varphi_{1_{yy}} - (2\varphi_{1_{xt}} + \varphi_{1_{tt}})/\epsilon = 0.$$

The ADI method of Borland, Rizzetta, and Yoshihara (ref. 17) was applied to the equation, and then the harmonic assumption for the nth approximation

$$\varphi_1^n (x,y,t) = \varphi_1^n (x,y)\, e^{-in\omega\Delta t}$$

was substituted into the difference equations. Here Δt is the time step between successive approximations, and has no physical significance to the problem. This yields a set of equations which has the same form as alternating row and column relaxation but for which the solution is convergent for all frequencies. The finite difference procedure was derived in reference 7 for two-dimensional flow. The equations for three-dimensional flow are also derived in Appendix A for the rectangular wing and in Appendix B for the swept and/or tapered wings.

5.0 ADI RELAXATION SOLUTIONS FOR AIRFOILS

In NASA CR 3537 (ref. 7), we showed that the direct solution of the difference equations for the harmonic motion of an airfoil in transonic small perturbation flow yields aerodynamic forces that take into account the effect of moving shock waves, at least in a first order sense. A program for two-dimensional flow based on the direct solution is reliable, efficient, and yields results which are in good agreement with experimental measurements (e.g., see ref. 7). Also, typical section flutter boundaries calculated with airforces from this program exhibit the characteristic 'transonic bucket.' The direct solution is also applicable to three-dimensional flow, but here the computer requirements are large and costly. Accordingly, a method of relaxation was developed in which simple harmonic motion was assumed after the finite difference formulation rather than before (see section 4.0). The resulting set of difference equations is solved by implicit alternating direction techniques, and hence the procedure has been entitled the "ADI" method. This ADI method does not have the convergence problem of the conventional block relaxation procedures. The differential equations were derived in NASA CR 3537 and preliminary two-dimensional examples were included. The derivations for three-dimensional rectangular wings and for swept, tapered wings are given in appendices A and B of this report.

The equations of the ADI procedure and direct solution differ in several respects. First, there is a time step, Δt, which appears in the ADI equations but not in the direct solution. This time step, which has no physical significance, does lead to truncation errors which are discussed in the next section. Second, the shock point operator which proved to be satisfactory for the direct solution had to be modified for use in the ADI difference equations. This is discussed in section 5.2.

To test the practicality of the method for eventual application to three dimensions, the two-dimensional ADI method was applied to a number of problems which include airfoil configurations both with and without thickness. Correlations are made with solutions from a program based on a compressible kernel function (see ref. 19), and from the direct solution program evaluated in reference 7.

Because of the low cost of each two-dimensional iteration, the solutions were generally run until the maximum of the difference between time steps was less than 10^{-4}, a criterion found acceptable for the earlier relaxation techniques. The number of iterations depends upon the time step chosen, smaller time steps requiring a proportional increase in the number of iterations. There also is a considerable Mach number influence on the rate of convergence. For a frequency of about 0.8 and choice of time step of 0.4, it was found that for $M = 0.9$ convergence required about 600 iterations while for $M = 0.7$ less than 300 were needed, with the variation with Mach number being roughly linear.

The following table gives data on the number of iterations required for convergence for several time steps for the severe case of $M = 0.9$ and a reduced frequency of $k = 0.9$ for a symmetric flow using a 72 by 60 mesh (2160 mesh points).

Time step	Iterations	CP seconds (CRAY)
0.1	1000	55
0.47	26	40
0.6	514	28
0.8	406	22
1.0	372	20

An indication of the superiority of the direct solutions for two dimensions is seen from the time required for the full (72 by 60) and half (72 by 30) matrix solutions; that is, 17 cps for the full matrix and 4.5 cps for the half matrix.

5.1 CORRELATION OF EXAMPLES FOR AIRFOILS OF VANISHING THICKNESS

The ADI method was first applied to calculating the pressure distribution on a flat plate pitching harmonically about the leading edge. Agreement between solutions from the ADI method and the program of reference 5.1 using an integral equation method were good for a Mach number of 0.9 and reduced frequencies up to 0.3 as seen in figure 1. However, agreement for a reduced frequency of 0.45 is not as good (fig. 2), and for a reduced frequency of 0.6 the agreement was found to be poor (see fig. 3). Reducing the time step improved the solution somewhat.

Basically, the time step is irrelevant to the problem but it does lead to truncation errors. By adding equations (A-5) and (A-6) to (A-7) and equating the three successive values of φ, that is, setting $\varphi^{\lambda} = \varphi^{n} = \varphi^{n+1}$, we obtain a difference equation which we can recognize as resulting in the following differential equation (see equations (A-62) and (A-64)):

$$(u\varphi_x)_x + \varphi_{yy} + \varphi_{zz} - \frac{4}{(1+\beta_1)}\left(\frac{1-\beta_1}{\epsilon\Delta t}\right)\varphi_x - \frac{2}{(1+\beta_1)}\left(\frac{1-\beta_1}{\Delta t}\right)^2 \frac{\varphi}{\epsilon} = 0$$

where $\beta_1 = e^{-i\omega\Delta t}$. When the time step Δt goes to zero with $u = K$, this equation reduces in the limit to the classical linearized equation for harmonic unsteady flow. To check whether the program contained basic errors, this differential equation (without the φ_{zz} term) was solved by the direct method used for the classical equation. The resulting solution agrees very closely with the ADI solution as seen in figure 4. This verifies that the algorithm was properly programmed. A second order φ_{tt} difference which required saving an additional time step was also tried but introduced no significant improvement.

Since the direct solution of the regular equations used a second order central difference for the φ_x term, a second order backward difference was derived and introduced into the ADI method. The derivation of this φ_x operator as well as the φ_{tt} operator enabling a change in the time step during a run are presented in Appendix D. Considerable improvement resulted, and now the ADI is in good agreement with the kernel function results. This is shown in figure 5 for a reduced frequency of 0.6. Therefore, to obtain sufficient accuracy at the higher frequencies we must use a second order difference for the first derivative in x.

The effect of varying the size of the time step, Δt, for $M = 0.9$ and $k = 0.6$ is shown in figures 6 and 7, with the results plotted with an expanded ordinate scale. Figure 6 presents solutions from the program of reference 19 and ADI results for the smallest and largest values of Δt tried. For the range studied, $\Delta t = 0.1$ to 1.5, the best correlation is obtained with the smallest value of Δt. Generally, the ADI calculations move towards the integral equation results in monotonic fashion as Δt is decreased. A reasonable compromise with accuracy for the sake of economy of calculations is found by using $\Delta t = 0.6$.

5.2 CORRELATION OF EXAMPLES FOR AIRFOILS WITH THICKNESS

A second set of examples was prepared for airfoils of finite thickness. The pressure coefficient distributions for the NACA 64A010 airfoil, oscillating in pitch about the quarter chord, tested by NASA Ames, were calculated with both the ADI and direct solution procedures. Calculations were made for a Mach number of $M = 0.85$ and reduced frequency of $k = 0.15$ and for $M = 0.86$ and $k = 0.4$. The resulting pressure distributions are presented in

figures 8 and 9. Agreement is quite good in both cases except for the real parts in the vicinity of the shock. In figure 8, the real part of the pressure aft of the shock from the ADI program is significantly larger than that from the direct solution. In figure 9, the ADI results underpredict the pulse while overpredicting the pressures aft of the shock. The difference in results between the ADI and direct solutions may well be a difference in shock point operators. Since use of the shock point operator which is the exact equivalent to that in the direct solution causes divergence in the ADI solution, the shock point operator was applied only to the second derivative with respect to x in the ADI program and not to both the first and second derivatives as in the direct solution.

Pressure distributions for a Mach number of 0.85 and for reduced frequencies of 0.25 and 0.4 are also presented in figures 10 and 11. The distributions are presented for two ADI solutions and the direct solution. The ADI solutions differ in the representation of the first derivative with respect to x, one using a first order finite difference and the other a second order. It is seen that there is relatively little difference between the two. However, in this particular case, the shock pulse of the ADI solutions differs considerably from that obtained by the direct solution.

Since a singularity occurs across the shock in the pressure distribution, a more accurate method of determining the influence of the shock point operator is to plot the difference in velocity potential across the airfoil. This jump is plotted in figure 12 for $k = 0.25$. Two ADI solutions are compared with the direct solution and are indicated by the legend on the figure. Again, one solution uses a first order difference for the first derivative with respect to x and the other uses a second order difference. For a shock of this strength, a distinct jump in the unsteady potential difference is observed, which can be related to the amplitude of shock motion by the shock relations derived in appendix A of reference 7. We note that there is very little jump in the real part of the potential difference at the shock for either ADI method compared with the direct solution.

To obtain a better shock representation, shock boundary conditions were derived for application to the ADI method of relaxation, and the derivation is presented in appendix D. The expressions are similar to those obtained for the direct solution and described in NASA CR 3537. An example of the captured shock on the NACA 64A010 at the Mach number of 0.85 is shown in figure 13. When the shock strength is defined as

$$[u] = u_{i+1j} - u_{ij}$$

some improvement in the jump in the real part of the unsteady potential for the airfoil oscillating in pitch about the leading edge is obtained, but results for the imaginary part are worse than those from the shock point operator (fig. 14). When the shock strength is defined by the more realistic value of

$$[u] = u_{i+2j} - u_{i-1j}$$

the jump in the real part of the unsteady potential difference is in close agreement with the direct solution, and the jump in the imaginary part is somewhat less than that for the direct solution. The corresponding pressure distributions are shown in figure 15. In the direct solution, shock boundary conditions were found to lead to somewhat different results for the jump in the unsteady potential from those obtained by the shock point operator. In reference 7, it was shown that the shock point operator for the direct solution lacked one term in the shock jump conditions. The application of the shock boundary conditions appears to be an acceptable method of representing the effect of the shock in unsteady flow in the ADI relaxation method.

To study the effect of first- and second-order difference representations of the φ_x term on airfoils with thickness, the preceding configuration was run at M = 0.85 and reduced frequencies of 0.25 and 0.40. The results of the ADI method for both representations of φ_x are compared with the direct solution in figures 10 and 11. The greatest deviation occurs in the vicinity of the shock where the pressure pulse is significantly underestimated by the ADI calculations. Inclusion of the second-order difference does not appear to be as important for the airfoil as for the configurations of vanishing thickness.

6.0 ADI RELAXATION SOLUTIONS FOR THREE-DIMENSIONAL WINGS

6.1 CORRELATION OF EXAMPLES FOR RECTANGULAR WINGS OF VANISHING THICKNESS

Pilot programs for three-dimensional unsteady transonic flow using the direct solution procedure (OPTRAN3) and the ADI relaxation method (OPTRAD3) have been applied to a zero thickness rectangular wing of aspect ratio 3 which corresponds to the planform geometry of the wing model of NASA TN D-344. Typical results from these two programs are compared with corresponding calculations from the RHOIV program of reference 8 in figures 16 through 18. These results are for a Mach number of 0.9 and a reduced frequency, based on the semichord, of 0.10, with the wing oscillating in pitch about the leading edge. The correlation of the results from the two finite difference procedures with the results from RHOIV is considered very good, even close to the tip.

For the real part, results from the three theories match nearly exactly for the inboard two chords, and RHOIV lies slightly below OPTRAD3 and OPTRAN3 for the tip chord. For the imaginary part, OPTRAD3 and RHOIV match closely over the inboard two chords with the OPTRAN3 curve lying slightly below. For the tip chord, OPTRAD3 and RHOIV match closely over the front part of the surface (OPTRAN3 again lies slightly below), while OPTRAN3 and RHOIV match closely aft of the midchord (OPTRAD3 lies slightly above). The better agreement of OPTRAD3 and RHOIV is surprising since truncation errors due to the time steps Δt were observed in the two-dimensional solutions.

The solutions are for a 50 by 20 by 40 xyz mesh (symmetry in z assumed so the order of the coefficient matrix is 20,000). As expected, the direct solution proved to be very expensive to run for this three-dimensional problem, and the ADI solution requires about a quarter of the computing resources of the direct solution.

6.2 CORRELATION OF EXAMPLES FOR RECTANGULAR WINGS OF FINITE THICKNESS

Steady-state pressure distributions for the wing of NASA TN D-344 with a 5%-thick circular arc airfoil are presented in figure 19. Included in the figures are pressure distributions for the upper and lower surfaces (as digitized from the graphs of TN D-344, ref. 20) and analytical results from XTRAN3S. It is noted that, despite a symmetric airfoil shape and zero angle of attack, the pressures for the upper and lower surfaces do not coincide. The report mentions a cross tunnel variation in stream angle which may account for the differences. The pressure distributions are presented for four spanwise stations. The analytical stations are closely matched to the experimental stations in all cases, so that a good estimate of the correlation between theory and experiment may be obtained by reviewing the figures. At the root (fig. 19), the analytical results extend to a larger negative pressure coefficient than the experimental results. Also, the analytical distribution shows a relatively sharp shock while the experimental results show some supersonic flow but almost no shock. Since there is a shock at mid semispan, a shock at the root would be expected too. The lack of a shock at the root may be due to the cross tunnel variation in stream angle and may also be due to boundary layer effects. The pressure orifices at the root are on the tunnel wall, rather than on the wing, and there is no mention of a splitter plate to remove the boundary layer on the tunnel wall. A thick boundary layer over the orifices might mask shock effects in the pressure measurements.

At the mid semispan section, $y/Ab = 0.5$ (fig. 19), the correlation between theory and experiment is quite good. The analytical result matches that for the upper surface well, particularly with respect to the shock strength. The maximum pressure is slightly less for the analytical result than for the experimental result, other-

wise the analytical result seems to lie between the measured distribution for the upper and lower surfaces. For y/Ab = 0.70 in figure 19, the analytical results fall slightly below the measured distributions, however, the shock strengths appear about the same. For y/Ab = 0.9 in figure 19, analytical results fall slightly farther below the measured distributions. Again, the shock strength looks about the same for analytical as for the measured results.

Using the potential distribution from the steady state solutions of XTRAN3S just discussed, we analyzed the wing of NASA TN D-344 (ref. 20), with its circular arc airfoil section, under the bending mode presented in figure 20. The results are for M = 0.9 and reduced frequency of 0.13. Three distributions are presented in each figure: (1) the ADI relaxation calculations for the wing with the circular arc airfoil, (2) the ADI calculations for a flat plate and (3) the experimental measurements as obtained from figure 9c of TN D-344 using an automated digitizing process on the unflagged data. The amplitude and phase angle of the pressure difference across the wing are presented for three spanwise stations: y/Ab = 0.5, 0.7, and 0.9.

The results from the ADI method in figures 21 and 22 for the thickness case reflect a much sharper and stronger pressure pulse than do the experimental measurements. Comparison of the amplitudes at the three spanwise stations is presented in figure 21. At y/Ab = 0.5 and 0.7, the experimental and analytical shock locations agree relatively well. At y/Ab = 0.9, ADI calculations show a very small shock pulse, while the presence of a shock in the experimental data is unclear due to variations between the flagged and unflagged data in figure 9c of TN D-344. Ahead of the shock, the amplitude of measured pressure is greater than the calculated pressure for all three spanwise stations.

A comparison of the phase angle distributions is given in figure 22. Overall, the correlation appears about like that for the pressure amplitudes. The analytical results show a sharp rise in phase angle across the shock at all three spanwise stations, with the phase angle decreasing aft of the shock for the two inboard stations. Although there are only two data points aft of the shock, the experimental data does generally resemble the analytic data. At y/Ab = 0.5 and 0.7 (fig. 22), the experimental data show a much larger jump in phase angle across the shock than the analytic data. Behavior of the experimental data aft of the shock, although described by just two data points, generally matches the analytic behavior at y/Ab = 0.9 in figure 22. The experimental data does show a large spike in phase angle (presumably due to a shock), while the analytic results show a jump across the shock.

The results of analyzing the wing of NASA TN D-344 presented in figures 21 and 22 are replotted in figures 23 and 24 to include the second set of experimental data (the flagged data) from the report. The flagged data were included in the report to show the repeatability of the measurements. Measurements in the vicinity of the shock show large variations between the two sets of data.

Three-dimensional plots of the real and imaginary parts of the calculated pressure distribution for this problem are shown in figures 25 and 26. Figure 25 is for the vanishing thickness configuration, while figure 26 is for the wing with a circular arc airfoil section. The pressure pulse due to the presence of the shock shows up dramatically for the circular arc configuration. There is a noticeable blip in the pressure distributions at the trailing edge of the tip chord for both the zero thickness and airfoil distributions. Such a blip has been encountered in other time-integration finite difference solutions. It seems to be associated with the solution mesh and is considered to make an insignificant contribution to the overall flow solution.

The overall correlation between theory and experiment is somewhat less than satisfying, although the basic properties of the pressure distribution in the experiments are reflected in the theoretical calculations. The

theoretical solution has been checked for convergence by running an additional 100 iterations with no apparent change in pressure distribution.

More recent experiments reported by Rickets, Sanford, Seidel, and Watson of the NASA Langley Research Center (ref. 21) correlate better with calculations by the frequency domain method. A rectangular wing of aspect ratio 4.0 with a supercritical 12%-thick airfoil is oscillated in pitch about the leading edge. Freon gas was used in the wind tunnel for which the ratio of specific heats is $\gamma = 1.131$.

The steady state solution for a Mach number of 0.7 and angle of attack of 2 deg. was computed by XTRAN3S. The resulting chordwise pressure coefficient distributions are shown in figures 27 through 30 at fractions of the semispan of $\bar{\eta} = 0.31, 0.59, 0.81$, and 0.95, respectively. The agreement of the experimental measurements with theory is seen to be quite good.

Using the steady-state potential just described for calculating the coefficients of the three-dimensional ADI method, we ran the program for the rectangular wing oscillating in pitch about the leading edge at reduced frequencies of 0.178 and 0.356. Figure 31 presents the distribution of amplitude of the jump in pressure across the wing for $k = 0.178$ at spanwise stations of $\bar{\eta} = 0.31, 0.59, 0.81$, and 0.95, while figure 32 shows the corresponding phase angle distributions. The pressure amplitude over the aft 3/4 chord of the wing is in good agreement with experimental values. The OPTRAD3 results show the leading edge singularity at all four spanwise stations, while this singularity appears only for the outer two chords in the experimental data. A shock pulse appears just aft of the leading edge for the two inboard chords in both the OPTRAD3 and the experimental results. The shock pulse is sharper in the theoretical calculations, and the theory shows a singularity at the leading edge which has the same sign at all cross sections. This is not seen in the experimental values. The agreement of the theory with the phase angle measurements is not as good as for the amplitude. The experimental results indicate a rise in phase angle to about 3/4 chord and then a fall to zero, while the theory continues upward. Similarly, comparison of the ADI results with the first harmonic from XTRAN3S, as digitized from the graphs of reference 21, is shown in figures 33 and 34. The pulse near the leading edge from XTRAN3S is stronger than the harmonic solution for $\bar{\eta} = 0.31$. For the more outboard spanwise locations the agreement is better. The phase angle from XTRAN3S shows a rapid rise at the trailing edge not observed either in the experiments or the ADI solution. For $k = 0.356$, the correlation of the ADI solution with experimental results is similar to that for $k = 0.178$, as seen in figures 35 and 36. The correlation with the first harmonic of XTRAN3S is shown in figures 37 and 38, and is similar to the results found for the correlation of two-dimensional results with XTRAN2.

6.3 SOLUTION CONVERGENCE PROBLEMS FOR SWEPT WINGS

The ADI program was modified to include the swept wing coordinate transformation. The problem of construction of a mesh system for a swept wing is considerably simplified by using a coordinate transformation which defines the leading and trailing edges as single values of the streamwise coordinate system. This makes it easier to refine the grid in the neighborhood of the leading and trailing edges.

Since XTRAN3S was selected to compute the steady-state potential distribution required in the coefficients of the difference equations, the mapping used in XTRAN3S was adapted to our program. The coordinate transformation is given by

$$\begin{aligned} \zeta &= (x - X_{LE})/S(y) - 1 \\ \eta &= y \\ \zeta &= z \end{aligned} \tag{6.31}$$

where S(y) is the semichord at the span location y. This differs from the transformation in reference 17 in order to make the leading and trailing edges be given by $\xi = -1$ and 1, respectively. The differential equation resulting from applying this transformation and the resulting difference equations are derived in appendix B.

The three-dimensional ADI relaxation program using the swept wing coordinate transformation has been applied to the clipped delta planform (as a zero thickness configuration) of Hess, Wynne, and Cazier (ref. 22), which has a swept leading edge of 50.45 deg. an unswept trailing edge, and a semispan of 0.708 chords. The time step had to be reduced to 0.002 before convergence occurred. However, after 570 iterations, the solution again started to diverge. A plot of the pressure distribution at the time when convergence stopped was in very poor agreement with the RHOIV solution. A modified clipped delta wing with a leading edge sweep angle of 20 deg. was also run to determine the effect of the leading edge sweep. The convergence rate was somewhat improved.

The grid region for the 50.45 deg. clipped delta wing is shown in figure 39. Note that the outboard section covers only a small region of the flow and may account in part, at least, for the lack of agreement with the RHOIV solution as well as the tendency toward solution divergence.

To find the effect of sweep alone, a simple swept untapered wing was analyzed. Convergence was obtained for a sweep angle of 20 deg. but failed at 30 deg. When the mapping was changed to make the leading and trailing edges turn normal to the line of symmetry as in figure 40, the convergence was extended to 30 deg. but a 40 deg. sweep failed to converge with the maximum growth occurring at the point where the edges curved sharply. At this point the mapping has discontinuous second derivatives. This suggests that a mapping with a continuous second derivative may improve the convergence. To test this, a swept untapered wing was analyzed using a cubic to bend the leading and trailing edges normal to the plane of symmetry. A finer grid spacing in the streamwise direction was also applied on the downstream boundary. It was found that, for the new transformation, convergence was obtained for sweep angles up to 45 degrees.

Additional studies concerning solution divergence of swept configurations using coordinate transformations may be found in references 23 and 24. In the former, care is taken to assure that the derivative ξ_y, which appears as a coefficient in the transformed differential equation, is well behaved in the solution region. In reference 24, linear stretching is applied to the transformation in the regions both upstream and downstream of the wing planform to assure that the region of perturbed flow is included in the flow solution region. Both concepts appear to significantly improve convergence. We assume that our new transformation is close to that used in reference 23. However, we have also made a point of removing the singularities resulting from the sharp planform apex at the root by artificially rounding off the leading and trailing edges at the root.

To test convergence at the higher frequencies, the 45 deg. sweep wing was calculated using $k = 0.5$. The solution converged as easily as for the lower frequency calculations. As seen from figures 50 through 52, the agreement with RHOIV is, however, not as good. As in the two dimensional solutions, the second order difference for the first derivative with respect to x appears to be required to improve the accuracy.

6.4 CORRELATION OF EXAMPLES FOR SWEPT WINGS OF VANISHING THICKNESS

Calculations were performed at a Mach number of 0.9 and reduced frequency of 0.13 for sweep angles of 30 deg, 40 deg, and 45 deg. Figures 41 through 43 show the real and imaginary parts of the jump in pressure coefficient across the flat plate for the three sweep angles using a time step of 0.05. The greatest variation with sweep angle occurs at the plane of symmetry (figure 41). The solutions for a sweep angle of 30 deg are shown in figures 44 through 46 for spanwise locations at fractions of semispan of 0., 0.51, and 0.93. The results after 150 and 300

iterations are compared with the integral equation solution of RHOIV. It appears that 300 iterations yield better results, and agreement with the RHOIV is seen to be excellent.

The jump in pressure coefficient for 45 deg, the largest sweep angle for which the program converged, is compared with the RHOIV solution in figures 47 through 49. The agreement is seen to be as good as for the lower sweep angles.

To test convergence at the higher frequencies, the 45-deg swept wing was calculated using $k = 0.5$. The solution converged as easily as for the lower frequency calculations. As seen from figures 50 through 52, the agreement with RHOIV is, however, not as good. As in the two dimensional solutions, the second-order difference for the first derivative with respect to x appears to be required to improve the accuracy.

7.0 CONJUGATE GRADIENT TECHNIQUES FOR THE DIRECT SOLUTION

7.1 DESCRIPTION OF BASIC CONJUGATE GRADIENT TECHNIQUES

Although the direct solution as used in OPTRAN2 provides an efficient and practical procedure for analyzing two-dimensional configurations with mesh systems of the order of 4500, three-dimensional problems using mesh configurations of the order of 50,000 to 100,000 would appear to require alternate procedures to achieve comparable efficiencies. Review of the literature indicated that conjugate gradient techniques might provide a practical way to obtain a direct solution for larger systems. However, our problem is asymmetric and indefinite, and thus the classical conjugate gradient methods are not applicable.

The classical conjugate gradient method for solving systems of linear equations of the form Ax = b whose coefficient matrices, A, are symmetric (Hermitian in the case of complex matrices) and positive definite was presented by Hestenes and Stiefel (ref. 25) in 1952. This method converges to the true solution of the linear system in a finite number of iterations in the absence of round-off errors and hence may be thought of as a direct solution. However, the method was not widely used and little was heard about it until the mid 1960s. J.K. Reid (ref. 26) noticed that the method is very efficient for large sparse symmetric positive definite systems that are well conditioned, with the asymptotic rate of convergence being inversely proportional to the square root of the condition number. In the last decade, many variants of the conjugate gradient method have been applied to large sparse problems with considerable success (See Hafez and Wong, ref. 27).

When the coefficient matrix, A, is asymmetric, the usual procedure is to multiply both sides of the equations by the conjugate transpose, A*, to obtain a Hermitian positive definite linear system of the form A*Ax = A*b on which to apply the conjugate gradient method. However, the condition number of the matrix A*A is the square of that of A. That is, the asymptotic convergence rate of the conjugate gradient method applied to the above equation would be inversely proportional to the condition number of A instead of to the square root of the condition number of A.

Various authors, such as Axelson (ref. 28), Concus and Golub (ref. 29), and Elman (ref. 30), have generalized the conjugate gradient method to a larger class of matrices, so that the asymptotic convergence rate is inversely proportional to the square root of the condition number of A. Yet all of these methods still assume the coefficient matrix to have certain properties, such as A* + A being positive definite, and thus they are not always applicable to general nonsingular matrices. In appendix E, a variant of the conjugate gradient method is developed which is applicable to general nonsingular matrices and which has an asymptotic convergence rate inversely proportional to the square root of the condition number. This new algorithm together with examples of its application will be discussed in the following.

However, for the new algorithm, when the coefficient matrix is even mildly ill-conditioned, the convergence rate of the conjugate gradient method is impractically slow. This difficulty, frequently encountered with the matrices for the transonic problem, can be remedied by preconditioning, which attempts to improve the condition number of the matrix. Various authors, such as Lewis and Rehm (ref. 31), Kershaw (ref. 32), and Chandra (ref. 33) have suggested different preconditioners for specific problems. The success of the method for a particular problem very much depends on the preconditioner chosen. For our problem we have tried preconditioners from the following sources:

(1) A sparse capacitance matrix method
(2) The ADI operator
(3) The incomplete LU factorization procedure.

The preconditioner from the incomplete LU factorization has proved to be most efficient. We shall describe these three preconditioners in detail in section 7.2.

7.2 PRINCIPLE OF PRECONDITIONING

The principle of preconditioning is to recast the system of equations in the form

$$(I + T)\, x = b$$

where T is a matrix whose magnitude is small. One procedure for accomplishing this is to find a matrix N such that its inverse is easy to compute and N is very close to the coefficient matrix A. If such an N can be found, then the matrix

$$R = A - N$$

is of small magnitude. The matrix $N^{-1}R$ is also of small magnitude if A and N are well conditioned. We can write the equation $Ax = b$ as $(N + R)\, x = b$. When we premultiply both sides of the latter equation by N^{-1}, then we obtain

$$(I - N^{-1}R)\, x = N^{-1}b$$

Note that the singular values of $(I + T) = (I - N^{-1}R)$ (See Householder, ref. 34) are defined as the square roots of the eigenvalues of

$$(I+T)^*(I+T) = I + T + T^* + T^*T.$$

Since the magnitude of T is small, the singular values of the matrix (I + T) are close to 1. This is the ideal situation for applying conjugate gradient type methods.

Aside from the case in which the matrix A is a so-called M-matrix, which is a matrix with negative diagonal entries, non-negative off-diagonal entries with all its eigenvalues on the right half-plane, (see Varga, ref. 35), there are no theoretical results in the literature concerning the choice of N. Almost all of the results in the literature concerning preconditioning are empirical and therefore are problem-dependent. The most popular and successful preconditioner is the incomplete factorization procedure.

7.2.1 THE INCOMPLETE FACTORIZATION

The concept of incomplete factorization may be used for preconditioning in the following manner: the L and U matrices resulting from the lower and upper triangular decompositions of the coefficient matrix (or some portions thereof) are modified to obtain matrices which are easy to invert and store. In the current algorithm, L and U are modified to be lower and upper tri-diagonal matrices.

The incomplete factorization of the matrix A in comparison with the complete factorization of A is illustrated in figures 53, 54, and 55. Figure 53 represents the sparsity of the original matrix A. Figure 54 shows the sparsity structure of the two matrices resulting from the complete LU factorization of matrix A. Figure 55 shows the sparsity structure of the incomplete LU factorization of the matrix A. The incomplete L and U are obtained from the regular Gaussian elimination by retaining only the nonzeroes in the sparsity structure of A. The details are expounded in appendix E. Our results are summarized as follows:

Problem	Grid size	No. of mesh points	Formulation	No. of iterations	CRAY CPU sec
(1)	72x60	2160	ADI (t = .3)	263	15.3
(2)	72x60	2160	direct	346	20.3
(3)	50x15x40	15000	direct	564	179.

In comparison with other methods, the ADI described in section 6 takes 40 seconds for problem 1, and the out-of-core solver takes 4.5 seconds for problem 2, and 400 seconds for problem 3.

In the two-dimensional problem, the ADI formulation (problem 1 of the test problems) seems to have better convergence properties than the direct solution formulation. This might well be true for the three-dimensional case also.

As an iterative method, the conjugate gradient is not so "operator-sensitive." For instance, in problem 1 of the test problems, second order central differencing is used for the φ_{xt} term. In the ADI method described in section 6, second order upwind differencing is used for the φ_{xt} terms because the ADI diverges when central differencing is used.

7.2.2 THE ADI OPERATOR AS A PRECONDITIONER

The ADI method can be represented by the following matrix equation:

$$\varphi_{n+1} = G\varphi_n + P\varphi_{n-1} + c \tag{7.2.1}$$

Assume equation (7.2.1) converges, then $\varphi_{n+1} = \varphi_n = \varphi_{n-1}$, and equation 7.2.1 3 can be written as

$$(I - G - P)\,\varphi = c. \tag{7.2.2}$$

We apply our conjugate gradient method (USYMLQ) to equation (7.2.1). Experimentally, we find that the rate of convergence of this method is similar to that of the ADI and hence more expensive than the ADI because for each iteration, we have to compute $(I - G - P)x_k$ as well as $(I - G - P)^*x_k$.

We have only tried this on the 2D problems. Since this method is not comparable in efficiency with the ADI, we do not list the results.

7.2.3 THE SPARSE CAPACITANCE MATRIX METHOD AS A PRECONDITIONER

The method we discuss in this section is motivated by the two-step method discussed in Ehlers and Weatherill (ref. 7). The flat plate equation without the boundary condition on the airfoil can be solved by Fast Poisson Solvers like FISHPACK (see Sweet and Swartztrauber, ref. 36). Since Ehlers and Weatherill (ref. 7) observed that the solution of the airfoil equation is close to the solution of the flat plate equation in the far field, we thought a matrix of the form:

$$M = \begin{bmatrix} S \\ F \end{bmatrix} \tag{7.2.3}$$

where S is the coefficient matrix of the difference equation of the airfoil equation for the near field mesh points (the mesh points inside the box which encloses the airfoil and all the supersonic points), and F is the coefficient matrix of the flat plate equation for the far field mesh points, would closely approximate the coefficient matrix of the airfoil equation (see fig. 56 below). In other words, if T is the coefficient matrix of the difference equation for the airfoil equation for the far field mesh points, that is, if

$$A = \begin{bmatrix} S \\ T \end{bmatrix}$$

we assume (T - F) is of small magnitude. Therefore we use M in equation (7.2.3) as a preconditioner. Equations of the form $M^{-1}x = b$ are solved by a sparse capacitance matrix method. This preconditioner proves to be inefficient for the following reasons:

1. The sparse capacitance matrix method requires an efficient complex sparse linear equation solver. The solver we were using, ME28A from Harwell, was written for IBM machines and is extremely inefficient for the CRAY.

2. The two-step method discussed in reference 7 proves to be successful only for symmetric flow, thus the matrix M in equation (7.2.3) does not closely approximate the original coefficient matrix. Therefore convergence is slow.

3. The box which encloses all the supersonic points can be almost as big as the full grid.

8.0 THE DIRECT SOLUTION FOR THREE-DIMENSIONAL WINGS

The pilot program for the direct solution of two-dimensional configurations has been rewritten for application to three-dimensional rectangular configurations. This new pilot program, called OPTRAN3, is programmed for the CRAY. Application of OPTRAN3 is discussed in section 8.1 and an improved algorithm for an out-of-core solver is discussed in section 8.2.

8.1 APPLICATION OF OUT-OF-CORE SOLVER

One large and several small examples were run to test OPTRAN3. The large run, the only one to approximate a practical configuration, was a 50 by 20 by 40 mesh system applied to an aspect ratio 3 rectangular wing of vanishing thickness. Since the flow for this configuration is symmetric top to bottom, this problem consisted of 20,000 points. Pressure distributions from this calculation are compared with results from OPTRAD3 and RHOIV in figures 16 through 18. Correlation of all the results from the two finite difference procedures with reference results from the kernel function method is good and is about what we would expect from our experience with two-dimensional calculations. The overwhelming characteristic for the example is its running time which was some 1200 CPU seconds. Perhaps of even more significance is the fact that using the company cost algorithm, the CPU seconds only accounted for about one-fourth of the total cost of the run, while the remainder of the cost is mainly due to input/output operations. It is also noted that costs rise rapidly with increasing mesh points. For example, a problem with three-fourths the mesh points requires about one-third as many CPU seconds.

The pilot program that was tested was developed on a version of the CRAY which has one million words of core storage. It was also developed during the installation period at Boeing of both the CRAY hardware and software. Now the operating system has been stabilized, and the original machine has been replaced by one with two million words of core storage. The larger memory allows us to bring larger blocks into memory. Larger blocks mean that the length of the vectorizable do-loops is longer, and execution time should be reduced significantly. It also means a reduction in the number of disk access operations. With the new algorithm, the number of words written on disk is also significantly reduced. When solutions for a number of right hand sides are required, the out-of-core solver may be competitive with iterative methods, such as the ADI and the preconditioned conjugate gradient procedure.

The total cost of the initial three-dimensional run using the direct solution program was large enough that it was decided to concentrate efforts on the ADI procedure of section 6. If the running time can be reduced significantly as described above, the advantage of being able to include a number of mode shapes (i.e., a number of right hand sides to the set of difference equations) with little increase in cost would make the direct solution again competitive with other procedures such as the ADI method.

8.2 AN IMPROVED OUT-OF-CORE SOLVER PROCEDURE

In the 2D program, the out-of-core direct solver, ETCSM, is a general-purpose banded solver. Because of the size of the 3D problem, we intend to design a direct solver specialized for our problem. The key idea in the new solver is "implicit factorization", which means that the LU factorization of the diagonal blocks, which are products of sparse matrices, is computed as needed. The "implicit factorization" requires the matrix to be block-tridiagonal. The following 3 by 3 block system serves to explain the difference between the algorithm used in ETCSM (the out-of-core solver of ref. 18) and implicit factorization:

We are interested in solving Ax = b, where A is represented by the block system:

$$A = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{8.2.1}$$

The matrix A is now decomposed into a product of block-lower and block-upper triangular matrices:

$$A = LU = \begin{bmatrix} I & 0 & 0 \\ L_{21} & I & 0 \\ 0 & L_{32} & I \end{bmatrix} * \begin{bmatrix} U_{11} & U_{12} & 0 \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix} \tag{8.2.2}$$

where, by comparison with equation (8.2.1), we see that

$$U_{11} = A_{11},\ U_{12} = A_{12}$$

$$L_{21} = A_{21}A_{11}^{-1},\ \ U_{22} = A_{22} - L_{21}U_{12},\ U_{23} = A_{23}$$

$$L_{32} = A_{32}U_{22}^{-1}$$

$$U_{33} = A_{33} - L_{32}U_{23}.$$

Thus Ax = b is to be solved in the form LUx = b. Let y = Ux. Then Ly = b can be solved by forward substitution:

$$\begin{aligned} y_1 &= b_1 \\ y_2 &= b_2 - L_{21}y_1 \\ y_3 &= b_3 - L_{32}y_2. \end{aligned} \tag{8.2.3}$$

The solution vector x, which satisfies the equation Ux = y, can now be found by backward substitution:

$$\begin{aligned} x_3 &= U_{33}^{-1}y_3 \\ x_2 &= U_{22}^{-1}(y_2 - U_{23}x_3) \\ x_1 &= U_{11}^{-1}(y_1 - U_{12}x_2) \end{aligned} \tag{8.2.4}$$

In practice we do not need to compute the inverses of U_{33}, U_{22}, and U_{11}, but can use their lower and upper triangular factors to obtain x_3, x_2, and x_1 by forward and backward substitution. Continued application of LU decomposition results in inverses being required only for diagonal matrices, a simplification that, in turn, requires storing only the original block matrices. Thus, for $L_{21} = A_{21}V_{11}^{-1}$, we store A_{21} and A_{11} (= U_{11}) instead of L_{21}.

In the explicit factorization, as in the algorithm used in ETCSM, all the nonzero blocks of the factorization (except the identity matrices of L) are stored on disk, and the sequence of calculation is exactly as those described by equations (8.2.2) to (8.2.4). In the implicit factorization, we store only the diagonal blocks U_{11}, U_{22}, and U_{33} in their factored form on disk in order to conserve disk space. The sequence of calculation for the factorization stage is the same as (8.2.2); the sequence of calculation for the forward substitution stage is different: in place of L_{21}, we use its equivalent, $A_{21}A_{11}^{-1}$; in place of L_{32}, we use its equivalent, $A_{32}U_{22}^{-1}$ (see equation (8.2.2)).

Thus equation (8.2.3) becomes:

$$\begin{aligned} y_1 &= b_1 \\ y_2 &= b_2 - A_{21}A_{11}^{-1}y_1 \\ y_3 &= b_3 - A_{32}A_{22}^{-1}y_2 \end{aligned} \tag{8.2.5}$$

The sequence of calculation for the backward substitution is the same as described by equation (8.2.4), with the exception that we take advantage of the fact that $U_{12} = A_{12}$, $U_{23} = A_{23}$, and substitute A_{12} in place of U_{12}, A_{23} in place of U_{23} in equation (8.2.4), which now becomes:

$$\begin{aligned} x_3 &= U_{33}^{-1}y_3 \\ x_2 &= U_{22}^{-1}(y_2 - A_{23}x_3) \\ x_1 &= U_{11}^{-1}(y_1 - A_{12}x_2) \end{aligned} \tag{8.2.6}$$

In this approach, we do not need to store the off-diagonal blocks of the lower triangular matrix L on disk, and we can obtain the off-diagonal blocks of the upper triangular matrix U readily from the original matrix A.

We have tried this approach on the two-dimensional problem; the cost is about one-third of that of ETCSM. For the 3D problem, the matrix needs to be reordered with one level of nested dissection to reduce the bandwidth in order to use this approach more effectively.

9.0 CONCLUSIONS

This investigation has centered on the development of procedures for calculating the unsteady transonic flow about harmonically oscillating three-dimensional wings. The work has included studies of the direct solution method, a relaxation procedure using ADI techniques, and preconditioned conjugate gradient procedures.

The most practical procedure for three-dimensional analyses requiring a limited number of mode shapes and using a computer with one million words of memory or less appears to be the ADI procedure. A pilot program, including a coordinate transformation for swept and tapered wings, has been developed and applied to several wings. Results appear generally reasonable and in good agreement with experiment and kernel function methods. Solution convergence characteristics appear similar to those of XTRAN3S, and results have been obtained for a swept, untapered wing with a leading edge sweep of 45 deg. However, it should be noted that the solution time is proportional to the number of modes.

A pilot program for rectangular wings using the direct solution has been developed for the CRAY computer. As expected, this program has proved expensive to run for practical problems. However, a significant part of this expense is due to the cost algorithm that penalized input/output operations. This was a problem with the one-million-word CRAY on which the current program was developed. With CRAYs having larger core storage now available (we are currently working with a two-million-word core, and there is talk of an eight- to thirty-two-million-word core capability in the future), the direct solution remains a viable procedure, particularly for problems involving a large number of modes, since solutions for additional modes are obtained at minimal extra cost.

Finally, an algorithm for the conjugate gradient procedure as applied to asymmetric, indefinite coefficient matrices has been developed with a solution convergence rate proportional to the square root of the condition number. Several preconditioning procedures have been tested, and one based on incomplete LU decomposition has proved most efficient. Running times appear to be generally comparable to those for the ADI procedure. This procedure is still under development.

APPENDIX A

DERIVATION OF THE ADI DIFFERENCE EQUATIONS FOR THE RECTANGULAR WING

A.1 THE BASIC TIME-DEPENDENT DIFFERENCE EQUATIONS

In the same manner as for the two dimensional ADI method, we consider the time-dependent linear equation for the perturbation unsteady potential. Thus

$$\left(\phi_{tt} + 2\phi_{xt}\right)/\epsilon = \left(u\phi_x\right)_x + \phi_{yy} + \phi_{zz} \tag{A-1}$$

Following the procedure used by Borland, Rizzetta, and Yoshihara (ref. 17), we write the difference equation for the three sweeps as

x sweep:

$$2\overleftarrow{\delta}_x \left(\phi^{\alpha}-\phi^n\right)/(\epsilon\,\Delta t) = \left[\delta_x\left(u\delta_x\phi^{\alpha}\right) + \delta_x\left(u\delta_x\phi^n\right)\right]/2 + \delta_{yy}\,\phi^n + \delta_{zz}\phi^n \tag{A-2}$$

y sweep:

$$2\overleftarrow{\delta}_x \left(\phi^{\lambda}-\phi^{\alpha}\right)/(\epsilon\,\Delta t) = \delta_{yy}\left(\phi^{\lambda}-\phi^n\right)/2 \tag{A-3}$$

z sweep:

$$\left(\phi^{n+1}-2\phi^n+\phi^{n-1}\right)/\left(\epsilon\,\Delta t^2\right) + 2\delta_x\left(\phi^{n+1}-\phi^{\lambda}\right)/(\epsilon\,\Delta t) = \delta_{zz}\left(\phi^{n+1}-\phi^n\right)/2 \tag{A-4}$$

The superscripts α and λ denote intermediate steps between the n and n + 1 iterations. The difference operator $\overleftarrow{\delta}_x$ is a backward difference, while $\delta_x(u\delta_x)$ is a central difference for u positive (elliptic) and backward difference for u negative (hyperbolic).

As in the two-dimensional version, we introduce harmonic motion and write for the nth and the next intermediate approximation

$$\phi^n = \varphi^n e^{i\,\omega\,n\,\Delta t},\ \phi^{\lambda} = \varphi^{\lambda} e^{i\,\omega\,(n+1)\,\Delta t},\ \phi^{\alpha} = \varphi^{\alpha} e^{i\,\omega\,(n+1)\,\Delta t}$$

Substituting these expressions into equations (A-2) through (A-4) yields

$$2\overleftarrow{\delta}_x \left(\varphi^{\alpha} - \beta_1\,\varphi^n\right)/(\epsilon\,\Delta t) = \delta_x\left(u\delta_x\varphi^{\alpha}\right)/2 + \beta_1\left[\delta_x\left(u\delta_x\varphi^n\right)/2 + \delta_{yy}\varphi^n + \delta_{zz}\varphi^n\right] \tag{A-5}$$

$$2\overleftarrow{\delta}_x\left(\varphi^{\lambda}-\varphi^{\alpha}\right)/(\epsilon\,\Delta t) = \delta_{yy}\left(\varphi^{\lambda}-\beta_1\varphi^n\right)/2 \tag{A-6}$$

$$\left(\varphi^{n+1} - 2\beta_1\varphi^n + \beta_1{}^2\varphi^{n-1}\right)/(\epsilon\,\Delta t) + 2\overleftarrow{\delta}_x\left(\varphi^{n+1} - \varphi^{\lambda}\right)/(\epsilon\,\Delta t) = \delta_{zz}\left(\varphi^{n+1} - \beta_1\varphi^n\right)/2 \tag{A-7}$$

where

$$\beta_1 = e^{-i\,\omega\,\Delta t}$$

To write the equation for coding, we use the same operators as in the direct two-dimensional method. For the difference operators, we write

$$2\overleftarrow{\delta}_x\varphi/(\epsilon\,\Delta t) = c_{3i}(\varphi_{ijk} - \varphi_{i-1\,jk})$$

$$\delta_x(u\delta_x\varphi) = 2c_i\,u_{i+1\,jk}(\varphi_{i+1\,jk} - \varphi_{ijk}) - 2d_i\,u_{ijk}(\varphi_{ijk} - \varphi_{i-1\,jk})$$

for $(u_{i+1\,jk} + u_{ijk}) > 0$,

$$\delta_x(u\delta_x\varphi) = 2c_{i-1}\,u_{ijk}\,(\varphi_{ijk} - \varphi_{i-1\,jk}) - 2d_{i-1}\,u_{i-1\,j\,k}(\varphi_{i-1\,jk} - \varphi_{i-2\,jk})$$

for $(u_{i+1\,jk} + u_{ijk}) < 0$.

$$\delta_{yy}\varphi = 2a_j(\varphi_{i\,j-1\,k} - \varphi_{ijk}) - 2b_j(\varphi_{ijk} - \varphi_{i\,j+1\,k})$$

$$\delta_{zz}\varphi = 2a_k(\varphi_{i\,jk-1} - \varphi_{ijk}) - 2b_k(\varphi_{ijk} - \varphi_{ij\,k+1}) \qquad \text{(A-8)}$$

where $C_{3i} = 2/\Delta\, t(x_i - x_{i-1})$, and the other coefficients are defined in reference 1.

A.2 THE X SWEEP DIFFERENCE EQUATIONS

With the aid of equations (A-8), equation (A-5) for the x sweep becomes

$$2\overleftarrow{\delta}_x\varphi^\alpha/(\epsilon\,\Delta t) - \delta_x\left(u\delta_x\varphi^\alpha\right)/2 = \beta_1\left[2\overleftarrow{\delta}_x\varphi^n/(\epsilon\,\Delta t) + \delta_x\left(u\delta_x\varphi^n\right)/2 + \delta_{yy}\varphi^n + \delta_{zz}\varphi^n\right] \qquad \text{(A-9)}$$

or

$$\begin{aligned} &c_{3i}\left(\varphi^\alpha_{ijk} - \varphi^\alpha_{i-1\,jk}\right) - c_i\,u_{i+1\,jk}\left(\varphi^\alpha_{i+1\,jk} - \varphi^\alpha_{ijk}\right) \\ &\quad + d_i u_{ijk}\left(\varphi^\alpha_{ijk} - \varphi^\alpha_{i-1\,jk}\right) = \beta_1\left[\,c_{3i}\left(\varphi^n_{ijk} - \varphi^n_{i-1\,jk}\right)\right. \\ &\quad + c_i u_{i+1\,jk}\left(\varphi^n_{i+1\,jk} - \varphi^n_{ijk}\right) - d_i u_{ijk}\left(\varphi^n_{ijk} - \varphi^n_{i-1\,jk}\right) \\ &\quad + 2a_j\left(\varphi^n_{i\,j-1\,k}\right) - \varphi^n_{ijk}\Big) - 2b_j\left(\varphi^n_{ijk} - \varphi^n_{i\,j+1\,k}\right) \\ &\quad \left. + 2a_k\left(\varphi^n_{ij\,k-1} - \varphi^n_{ijk}\right) - 2b_k\left(\varphi^n_{ijk} - \varphi^n_{ij\,k+1}\right)\right] \end{aligned} \qquad \text{(A-10)}$$

Writing the equation in the form

$$\mathrm{SUB1(I)} * \varphi^\alpha_{i-2\,jk} + \mathrm{SUB(I)} * \varphi^\alpha_{i-1\,jk} + \mathrm{DIAG(I)} * \varphi^\alpha_{ijk} + \mathrm{SUPER(I)} * \varphi^\alpha_{i+1\,jk} = \mathrm{RHS\ (I)} \qquad \text{(A-11)}$$

we see that

$$\text{SUB1(I)} = 0.0$$

$$\text{SUB(I)} = -c_{3i} - d_i u_{ijk}$$

$$\text{DIAG(I)} = c_{3i} + c_i u_{i+1\,jk} + d_i u_{ijk}$$

$$\text{SUPER(I)} = -c_i u_{i+1\,jk} \tag{A-12}$$

Note that DIAG(I) = – SUB(I) – SUPER(I). The right hand side term becomes

$$\text{RHS(I)} = \beta_1(\text{RHS1} + \text{RHS2} + 2*\text{RHS3} + 2*\text{RHS4}) + 2\,\text{RHS4}) \tag{A-13}$$

where

$$\text{RHS1} = c_{3i}\left(\varphi^n_{ijk} - \varphi^n_{i-1\,jk}\right)$$

$$\text{RHS2} = c_i u_{i+1\,jk}\left(\varphi^n_{i+1\,jk} - \varphi^n_{ijk}\right) - d_i u_{ijk}\left(\varphi^n_{ijk} - \varphi^n_{i-1\,jk}\right)$$

$$\text{RHS3} = a_j\left(\varphi^n_{i\,j-1\,k} - \varphi^n_{ijk}\right) - b_j\left(\varphi^n_{ijk} - \varphi^n_{i\,j+1\,k}\right)$$

$$\text{RHS4} = a_k\left(\varphi^n_{ij\,k-1} - \varphi^n_{ijk}\right) - b_k\left(\varphi^n_{ijk} - \varphi^n_{ij\,k+1}\right) \tag{A-14}$$

When the point ijk is supersonic, then equation (A-10) becomes

$$c_{3i}\left(\varphi^\alpha_{ijk} - \varphi^\alpha_{i-1\,jk}\right) - c_{i-1}u_{ijk}\left(\varphi^\alpha_{ijk} - \varphi^\alpha_{i-1\,jk}\right) + d_{i-1}u_{i-1\,jk}\left(\varphi^\alpha_{i-1\,jk} - \varphi^\alpha_{i-2\,jk}\right) =$$

$$\beta_1\left[c_{3i}\left(\varphi^n_{ijk} - \varphi^n_{i-1\,jk}\right) + c_{i-1}u_{ijk}\left(\varphi^n_{ijk} - \varphi^n_{i-1\,jk}\right) - d_{i-1}u_{i-1\,jk}\left(\varphi^n_{i-1\,jk} - \varphi^n_{i-2\,jk}\right)\right.$$

$$+ 2a_j\left(\varphi^n_{i\,j-1\,k} - \varphi^n_{ijk}\right) - 2b_j\left(\varphi^n_{ijk} - \varphi^n_{i\,j+1\,k}\right)$$

$$\left. + 2a_k\left(\varphi^n_{ij\,k-1} - \varphi^n_{ijk}\right) - 2b_k\left(\varphi^n_{ijk} - \varphi^n_{ij\,k+1}\right)\right] \tag{A-15}$$

Comparison of equation (A-15) with equation (A-11) yields

$$\text{SUB1(I)} = -d_{i-1}u_{i-1\,jk}$$

$$\text{SUB(I)} = -c_{3i} + c_{i-1}u_{ijk} + d_{i-1}u_{i-1\,jk}$$

$$\text{DIAG(I)} = c_{3i} - c_{i-1}u_{ijk}$$

$$\text{SUPER (I)} = 0.0 \tag{A-16}$$

Note that SUB(I) = − SUB1(I) − DIAG(I). The terms RHS1 and RHS2 become

$$RHS1 = c_{3i}\left(\varphi^{n}_{ijk} - \varphi^{n}_{i-1\,jk}\right)$$

$$RHS2 = c_{i-1}u_{ijk}\left(\varphi^{n}_{ijk} - \varphi^{n}_{i-1\,jk}\right) - d_{i-1}u_{i-1\,jk}\left(\varphi^{n}_{i-1\,jk} - \varphi^{n}_{i-2\,j\,k}\right) \qquad \text{(A-17)}$$

RHS3 and RHS4 remain unchanged.

A.3 BOUNDARY CONDITIONS FOR THE X SWEEP

We apply the outgoing wave type boundary conditions of Engquist and Majda (ref. 11) on the upstream and downstream boundaries. Following the two-dimensional method, we obtain

$$\varphi^{\alpha}_{1jk} = \overline{C}_{k1}\varphi^{\alpha}_{2jk} + \beta_1\left(\varphi^{n}_{2jk} - \overline{C}_{k1}\varphi^{n}_{1jk}\right) \qquad \text{(A-18)}$$

$$\varphi^{\alpha}_{i_{max}jk} = \overline{C}_{k3}\varphi^{\alpha}_{i_{max}-1\,jk} + \beta_1\left(\varphi^{n}_{i_{max}-1\,jk} - \overline{C}_{k3}\varphi^{n}_{i_{max}jk}\right) \qquad \text{(A-19)}$$

where

$$C_{k1} = M(x_2 - x_1)/[(1-M)\,\Delta t]$$

$$C_{k3} = M\left(x_{i_{max}} - x_{i_{max}-1}\right)/[(1+M)\,\Delta t]$$

$$\overline{C}_{k1} = (1 - C_{k1})/(1 + C_{k1})$$

$$\overline{C}_{k3} = (1 - C_{k3})/(1 + C_{k3}) \qquad \text{(A-20)}$$

The equations (A-12) are modified for i = 2 and i = i_{max}-1 by

$$DIAG(2) + DIAG(2) + SUB(2)*\overline{C}_{k1}$$

$$RHS(2) = RHS(2) - SUB(2)*\beta_1\left(\varphi^{n}_{2jk} - \overline{C}_{k1}\,\varphi^{n}_{1jk}\right)$$

$$SUB(2) = 0.0 \qquad \text{(A-21)}$$

$$DIAG(IMAX1) = DIAG(IMAX1) + SUPER\,(IMAX1)*\overline{C}_{k3}$$

$$RHS(IMAX1) = RHS(IMAX1) - SUPER(IMAX1)*\beta_1\left(\varphi^{n}_{i_{max}-1\,j\,k} - \overline{C}_{k3}\varphi^{n}_{i_{max}jk}\right)$$

$$SUPER(IMAX1) = 0.0 \qquad \text{(A-22)}$$

For $k = k_m$, just below the wing plane, and $k = k_m + 1$, we apply the wing boundary conditions when $i_1 \leq i \leq i_0$, for $j < j_s$, the index of the first spanwise variable beyond the wing tip. The conditions are in the form

$$(\partial \varphi / \partial z)_{z \to -0} = F^L(x,y)$$

$$(\partial \varphi / \partial z)_{z \to +0} = F^U(x,y)$$

For $k = k_m$, we have

$$b_{k_m} (\varphi_{ijk_m} - \varphi_{ij\, k_m+1}) = - hF^L_{ij} b_{k_m} \tag{A-23}$$

where $h = z_{k_m+1} - z_{k_m}$. Similarly, for $k = k_m + 1$,

$$a_{k_m+1} (\varphi_{ijk_m} - \varphi_{ij\, k_m+1}) = - ha_{k_m+1} F^U_{ij} \tag{A-24}$$

We see that for $k = k_m$, the right hand side term is modified by

$$RHS(I) = RHS(I) + 2\beta_1 b_{k_m} \left[hF^L_{ij} + \varphi^n_{ijk_m} - \varphi^n_{ij\, k_m+1} \right] \tag{A-25}$$

Similarly, for $k = k_m + 1$ we have

$$RHS(I) = RHS(I) - 2a_{k_m+1} \beta_1 \left[hF^U_{ij} + \varphi^n_{ijk_m} - \varphi^n_{ij\, k_m+1} \right] \tag{A-26}$$

For $i > i_1$ and $j < j_s$, we must satisfy the continuity of the normal derivative across the wake sheet and continuity of pressure. Detailed discussion of this is given in reference 1. The jump in the potential at $x = x_{i_1} + 1$ is chosen to satisfy the Kutta condition that the jump in pressure at the trailing edge be zero. Continuity of pressure is assumed by setting

$$\Delta\varphi_{ij} = \Delta\varphi_{i_1+1\, j}\, e^{-i\omega (x_i - x_{i_1+1})}$$

where $\Delta\varphi_{ij}$ denotes the jump in potential at $x = x_i$. The continuity of normal derivative is assured by adding to the φ_{zz} term the quantity

$$- b_{k_m} \Delta\varphi_{ij}$$

for $k = k_m$ and

$$a_{k_m+1} \Delta\varphi_{ij}$$

for $k = k_m + 1$.

A.4 THE DIFFERENCE EQUATIONS FOR THE Y SWEEP

The difference equation for the y sweep, equation (A-6), becomes

$$- 2\overleftarrow{\delta}_x\varphi^{\lambda}/(\epsilon\,\Delta t) + \delta_{yy}\varphi^{\lambda}/2 = \beta_1\delta_{yy}\varphi^{n}/2 - 2\overleftarrow{\delta}_x\varphi^{\alpha}/(\epsilon\,\Delta t) \tag{A-27}$$

or

$$\begin{aligned} a_j\left(\varphi^{\lambda}_{i\,j-1\,k} - \varphi^{\lambda}_{ijk}\right) - b_j\left(\varphi^{\lambda}_{ijk} - \varphi^{\lambda}_{i\,j+1\,k}\right) - c_{3i}\left(\varphi^{\lambda}_{ijk} - \varphi^{\lambda}_{i-1\,jk}\right) \\ = \beta_1\left[a_j\left(\varphi^{n}_{i\,j-1\,k} - \varphi^{n}_{ijk}\right) - b_j\left(\varphi^{n}_{ijk} - \varphi^{n}_{i\,j+1\,k}\right)\right] \\ - c_{3i}\left(\varphi^{\alpha}_{ijk} - \varphi^{\alpha}_{i-1\,jk}\right) \end{aligned} \tag{A-28}$$

Writing this equation as

$$\mathrm{SUB(J)} * \varphi^{\lambda}_{i\,j-1\,k} + \mathrm{DIAG(J)} * \varphi^{\lambda}_{ijk} + \mathrm{SUPER(J)} * \varphi^{\lambda}_{i\,j+1\,k} = \mathrm{RHS(J)} \tag{A-29}$$

and comparing equations (A-28) and (A-29), we obtain

$$\begin{aligned} \mathrm{SUB(J)} &= a_j \\ \mathrm{DIAG(J)} &= - a_j - b_j - c_{3i} \\ \mathrm{SUPER(J)} &= b_j \\ \mathrm{RHS(J)} &= \beta_1\left[a_j\left(\varphi^{n}_{i\,j-1\,k} - \varphi^{n}_{ijk}\right) - b_j\left(\varphi^{n}_{ijk} - \varphi^{n}_{i\,j+1\,k}\right)\right] \\ &\quad - c_{3i}\left(\varphi^{\alpha}_{ijk} - \varphi^{\alpha}_{i-1\,jk} + \varphi^{\lambda}_{i-1\,jk}\right) \end{aligned} \tag{A-30}$$

Note that $\varphi^{\lambda}_{i-1\,j\,k}$ is known from the previous sweep, since we always move in increasing i. Since the φ^{λ} and φ^{α} values are not saved, the quantity

$$\varphi^{\alpha}_{i-1\,jk}$$

has been written over with $\varphi^{\lambda}_{i-1\,jk}$ by the previous step in the y sweep. Hence, after the i - 1 sweep we must save $\varphi^{\lambda}_{i-1\,jk}$ by

$$\mathrm{PHIOLD(J)} = \varphi^{\alpha}_{i-1\,jk} \tag{A-31}$$

before storing the results of the i - 1 step. In each step of the y sweep the value of k is fixed, while i varies from i = 2 to i_{max} - 1. Then k is increased by 1 and i is varied from 2 to i_{max} - 1 until the entire grid is swept.

A.5 BOUNDARY CONDITIONS FOR THE Y SWEEP

The difference equations for the y sweep require boundary conditions on the upstream x boundary. Hence, the value of i = 2 is a special case. Then the term $-c_{3i}\varphi_{i-1\,jk}$ is missing from the right hand side RHS(J) term, and we replace it with the expression

$$c_{3i}\,\overline{C}_{k1}\left[\varphi^{\alpha}_{2jk} - \varphi^{\lambda}_{2jk}\right]$$

The diagonal term is modified by

$$\text{DIAG(J)} = \text{DIAG(J)} + c_{32}\overline{C}_{k1}$$

The line y = 0 (j = 2) is a plane of symmetry for the solution since it is the location of the root chord. Thus $\delta\varphi/\delta y = 0$ at y = 0 yields

$$\varphi^{\lambda}_{i1k} = \varphi^{\lambda}_{i3k} \tag{A-32}$$

Application of the boundary condition to equation (A-29) yields

$$\begin{aligned}&\text{SUPER(2)} = \text{SUPER(2)} + \text{SUB(2)}\\ &\text{SUB(2)} = 0.0\end{aligned} \tag{A-33}$$

We apply the Majda and Engquist nonreflecting boundary conditions at the outer spanwise boundary, $y = (y_{j\,max} + y_{jmax\,-\,1})/2$. This takes the form

$$\varphi^{\lambda}_{i\,j_{max}k} = \overline{C}_{k2}\varphi^{\lambda}_{i\,j_{max}-1\,k} + \beta_1\left(\varphi^{n}_{i\,j_{max}-1\,k} - \overline{C}_{k2}\varphi^{n}_{i\,j_{max}\,k}\right) \tag{A-34}$$

where

$$\begin{aligned}\overline{C}_{k2} &= (1 - C_{k2})/(1 + C_{k2})\\ C_{k2} &= M\left(y_{j_{max}} - y_{j_{max}-1}\right)\sqrt{K}/\left(1 - M^2\right)\Delta t\\ K &= \left(1 - M^2\right)/\left(M^2\epsilon\right)\end{aligned} \tag{A-35}$$

This requires the following modifications

$$\begin{aligned}&\text{DIAG(JMAX1)} = \text{DIAG(JMAX1)} + C_{k2}\text{*SUPER(JMAX1)}\\ &\text{RHS(JMAX1)} = \text{RHS(JMAX1)} - \text{SUPER(JMAX1)} * \beta_1\left(\varphi^{n}_{ij_{max}-1\,k} - \varphi^{n}_{ij_{max}k}\right)\\ &\text{SUPER(JMAX1)} = 0.0\end{aligned} \tag{A-36}$$

A.6 THE DIFFERENCE EQUATIONS FOR THE Z SWEEP

The difference equation for the z sweep in equation (A-7) may be written as

$$\delta_{zz}\varphi^{n+1}/2 - \varphi^{n+1}/(\epsilon\,\Delta t^2) - 2\overleftarrow{\delta}_x\varphi^{n+1}/(\epsilon\,\Delta t)$$

$$= \beta_1\left[\delta_{zz}\varphi^{n}/2 - \left(2\varphi^{n} - \beta_1\varphi^{n-1}\right)/(\epsilon\Delta t^2)\right] - 2\overleftarrow{\delta}_x\varphi^{\lambda}/(\epsilon\,\Delta t) \qquad \text{(A-37)}$$

With the aid of equation (A-8) we obtain

$$a_k\left(\varphi^{n+1}_{ij\,k-1} - \varphi^{n+1}_{ijk}\right) - b_k\left(\varphi^{n+1}_{ijk} - \varphi^{n+1}_{ij\,k+1}\right) - E_1\varphi^{n+1}_{ijk} - c_{3i}\left(\varphi^{n+1}_{ijk} - \varphi^{n+1}_{i-1\,jk}\right)$$

$$= \beta_1\left[a_k\left(\varphi^{n}_{ij\,k-1} - \varphi^{n}_{ijk}\right) - b_k\left(\varphi^{n}_{ijk} - \varphi^{n}_{ij\,k+1}\right)\right.$$

$$\left. - E_1\left(2\varphi^{n}_{ijk} - \varphi^{n-1}_{ijk}\beta_1\right)\right] \qquad \text{(A-38)}$$

Writing equation (A-38) in the form

$$\text{SUB(K)} * \varphi^{n+1}_{ij\,k-1} + \text{DIAG(K)} * \varphi^{n+1}_{ijk} + \text{SUPER(K)} * \varphi^{n+1}_{ij\,k+1} = \text{RHS(K)} \qquad \text{(A-39)}$$

and comparing with equation (A-38) yields the following relations for the coefficients:

$$\text{SUB(K)} = a_k$$

$$\text{DIAG(K)} = -a_k - b_k - E_1 - c_{3i}$$

$$\text{SUPER(K)} = b_k$$

$$\text{RHS(K)} = \beta_1\left[a_k\left(\varphi^{n}_{ij\,k-1} - \varphi^{n}_{ijk}\right) - b_k\left(\varphi^{n}_{ijk} - \varphi^{n}_{ij\,k+1}\right)\right.$$

$$\left. - E_1\left(2\varphi^{n}_{ijk} - \beta_1\varphi^{n-1}_{ijk}\right)\right] - c_{3i}\left(\varphi^{\lambda}_{ijk} - \varphi^{\lambda}_{i-1\,jk} + \varphi^{\alpha}_{i-1\,jk}\right) \qquad \text{(A-40)}$$

where $E_1 = 1/\varepsilon\Delta t^2$. Note that the right hand side term, RHS(K), contains the known value of the n + 1 approximation, $\varphi^{n+1}_{i-1\,j\,k}$. Since the n + 1 approximation replaces the λ approximation, $\varphi^{\lambda}_{i-1\,j\,k}$ must be saved in the previous step. Thus we define

$$\text{PHIOLD(K)} = \varphi^{\lambda}_{i-1\,jk} \qquad \text{(A-41)}$$

A.7 BOUNDARY CONDITIONS FOR THE Z SWEEP

For the z sweep we must apply the boundary conditions on the wing, on the wake, and on the mesh boundaries except for the side boundaries, that is, the x-z plane boundaries. On the upstream boundary, we use

$$\varphi^{n+1}_{1jk} = \overline{C}_{k1}\varphi^{n+1}_{2jk} + \beta_1\left(\varphi^{n}_{2jk} - \overline{C}_{k1}\varphi^{n}_{1jk}\right) \tag{A-42}$$

No boundary conditions are actually needed in the difference equations on the downstream boundary, but for the next step we need the values of φ_1 which satisfy the downstream boundary conditions. Thus we set

$$\varphi^{n+1}_{i_{max}jk} = \overline{C}_{k3}\varphi^{n+1}_{i_{max}\text{-}1\ jk} + \beta_1\left(\varphi^{n}_{i_{max}\text{-}1\ jk} - \overline{C}_{k3}\varphi^{n}_{i_{max}\ jk}\right) \tag{A-43}$$

where

$$C_{k3} = M\left(x_{i_{max}} - x_{i_{max}\text{-}1}\right)/(M+1)\,\Delta t$$

$$\overline{C}_{k3} = (1 - C_{k3})/(1 + C_{k3}) \tag{A-44}$$

On the upper boundary, we have for the nonreflecting boundary conditions

$$\varphi^{n+1}_{ij\ k_{max}} = \overline{C}_{k5}\varphi^{n+1}_{ij\ k_{max}\text{-}1} + \beta_1\left(\varphi^{n}_{ij\ k_{max}\text{-}1} - \overline{C}_{k5}\varphi^{n}_{ij\ k_{max}}\right) \tag{A-45}$$

where

$$C_{k5} = M\left(z_{k_{max}} - z_{k_{max}\text{-}1}\right)\sqrt{K}/\left[(1 - M^2)\Delta t\right]$$

$$\overline{C}_{k5} = (1 - C_{k5})/(1 + C_{k5}) \tag{A-46}$$

The lower boundary nonreflecting condition is

$$\varphi^{n+1}_{ij1} = \overline{C}_{k4}\varphi^{n+1}_{ij2} + \beta_1\left(\varphi^{n}_{ij2} - \overline{C}_{k4}\varphi^{n}_{ij1}\right) \tag{A-47}$$

where

$$C_{k4} = M(z_2 - z_1)\sqrt{K}/(1 - M^2)\Delta t$$

$$\overline{C}_{k4} = (1 - C_{k4})/(1 + C_{k4}) \tag{A-48}$$

On the wing, for $i_0 \le i \le i_1$ where i_0 and i_1 may depend on j, we have for $z = (z_{k_m} + z_{k_m+1})/2 = 0$,

$$(\partial\varphi/\partial z)^{-} = F^{L}_{ij}$$

$$(\partial\varphi/\partial z)^{+} = F^{U}_{ij}$$

For $k = k_m$, the term $b_k(\varphi^{n+1}_{ijk} - \varphi^{n+1}_{ijk+1})$ becomes

$$b_{k_m}\left(\varphi^{n+1}_{ijk_m} - \varphi^{n+1}_{ij\,k_m+1}\right) = -b_{k_m} h \frac{\partial \varphi}{\partial z} = -b_{k_m} h F^L_{ij} \tag{A-49}$$

For $k = k_m + 1$, the term $a_{km}(\varphi^{n+1}_{ijk-1} - \varphi^{n+1}_{ijk})$ becomes

$$a_{k_m+1}\left(\varphi^{n+1}_{ijk_m} - \varphi^{n+1}_{ij\,k_m+1}\right) = -h\, a_{k_m+1} F^U_{ij} \tag{A-50}$$

These same boundary conditions must be applied to the corresponding terms for φ^n as well. On the wake for $i > i_1$, to satisfy the continuity of the normal derivative across the wake, we must add for $k = k_m$ the terms

$$b_{k_m}\left(\varphi^{n+1}_{ij\,k_m} - \varphi^{n+1}_{ij\,k_m+1} - \varphi^{n+1}_L + \varphi^{n+1}_U\right) = b_{k_m}\left(\varphi^{n+1}_{ijk_m} - \varphi^{n+1}_{ijk_m+1}\right) + b_{k_m}\Delta\varphi^{n+1}_{ij} \tag{A-51}$$

where $\Delta\varphi = \varphi_U - \varphi_L$. Similarly, for $k = k_m + 1$ we have

$$a_{k_m+1}\left(\varphi^{n+1}_{ij\,k_m} - \varphi^{n+1}_{ij\,k_m+1}\right) + a_{k_m+1}\Delta\varphi^{n+1}_{ij} \tag{A-52}$$

Equations (A-51) and (A-52) are applied to φ^n as well.

For all k, the boundary conditions on the upstream boundary require the following modifications of the coefficients. When $i = 2$,

$$\begin{aligned} &\text{DIAG(K)} = \text{DIAG(K)} + c_{32}\,\overline{C}_{k1} \\ &\text{RHS(K)} = \text{RHS(K)} - c_{32}\,\beta_1\left(\varphi^n_{2jk} - \varphi^n_{1jk}\,\overline{C}_{k1}\right) \end{aligned} \tag{A-53}$$

On the lower boundary, for $k = 2$

$$\begin{aligned} &\text{DIAG(2)} = \text{DIAG(2)} + \overline{C}_{k4} * \text{SUB(2)} \\ &\text{RHS(2)} = \text{RHS(2)} - \text{SUB(2)} * \beta_1\left(\varphi^n_{ij2} - \overline{C}_k\,\varphi^n_{ij1}\right) \\ &\text{SUB(2)} = 0.0 \end{aligned} \tag{A-54}$$

On the upper boundary, for $k = k_{max} - 1 = \text{KMAX1}$

$$\begin{aligned} &\text{DIAG(KMAX1)} = \text{DIAG(KMAX1)} + \text{SUPER(KMAX1)} * \overline{C}_{k5} \\ &\text{RHS(KMAX1)} = \text{RHS(KMAX1)} - \text{SUPER(KMAX1)} * \beta_1\left(\varphi^n_{ij\,k_{max}-1} - \overline{C}_{k5}\varphi^n_{ij\,k_{max}}\right) \\ &\text{SUPER(KMAX1)} = 0.0 \end{aligned} \tag{A-55}$$

For the wing boundary conditions at $k = k_m$, we have

$$DIAG(KM) = DIAG(KM) + b_{k_m}$$

$$SUPER(K) = 0.0$$

$$RHS(KM) = RHS(KM) - b_{k_m} h F^{L}_{ij} + \beta_1 \left[b_{k_m} \left(\varphi^{n}_{ij\, k_m} - \varphi^{n}_{ij\, k_m+1} \right) + b_{k_m} h F^{L}_{ij} \right] \quad \text{(A-56)}$$

For the wing boundary conditions at $k = k_m + 1$, we have

$$DIAG(KMP) = DIAG(KMP) + a_{k_m+1}$$

$$SUB(KMP) = 0.0$$

$$RHS(KMP) = RHS(KMP) - \beta_1 \left[h\, a_{k_m+1} F^{U}_{ij} + a_{k_m+1} \left(\varphi^{n}_{ijk_m} - \varphi^{n}_{ijk_m+1} \right) \right] + a_{k_m+1} h F^{U}_{ij} \quad \text{(A-57)}$$

For the wake boundary conditions, at $k = k_m$ we have

$$RHS(KM) = RHS(KM) + b_{k_m} \left(\Delta\varphi^{n+1}_{ij} - \beta_1 \Delta\varphi^{n}_{ij} \right)$$

$$RHS(KMP) = RHS(KMP) - a_{k_m+1} \left(\Delta\varphi^{n+1}_{ij} - \beta_1 \Delta\varphi^{n}_{ij} \right) \quad \text{(A-58)}$$

A.8 PROVISION FOR THE SYMMETRY OF THE STEADY FLOW

When the steady flow is symmetric and

$$F^{U}_{ij} = F^{L}_{ij} \quad \text{(A-59)}$$

then the unsteady potential is antisymmetric, and we have

$$\varphi_{ijk_m+1} = - \varphi_{ijk_m}$$

$$\varphi_{ijk_m+2} = - \varphi_{ijk_m-1} \quad \text{(A-60)}$$

These may be employed as boundary conditions, and half the matrix can be solved with a considerable saving in computing cost.

In place of the boundary conditions for $K = KMAX1\ (k_{max}-1)$, we apply boundary conditions for $k = k_m$. Thus the asymmetry conditions yield for $k = k_m$ and for $i \geq i_0$ and $i \leq i_1$,

$$DIAG(KM) = DIAG(KM) - b_{k_m}$$

$$SUPER(KM) = 0.0 \quad \text{(A-61)}$$

A.9 EXACT EQUATION SOLVED BY THE ADI METHOD

The ADI method of relaxation leads to a complex potential in much the same way as the block relaxation procedure that is used to solve the harmonic oscillation equation. The time step Δt is irrelevant to the solution, but its use leads to truncation errors which are of the order of Δt. To find the differential equation solved by the ADI method, we add equations (A-5), (A-6), and (A-7). We obtain

$$\left(\varphi^{n+1} - 2\beta_1\varphi^n + \beta_1^2\varphi^{n-1}\right)/\left(\epsilon\,\Delta t^2\right) - 2\overleftarrow{\delta}_x\left(\varphi^{n+1} - \beta_1\varphi^n\right)/(\epsilon\,\Delta t)$$

$$= \delta_x\left(u\delta_x\varphi^{\alpha}\right)/2 + \beta_1\delta_x\left(u\delta_x\varphi^n\right)/2 + \delta_{yy}\left(\varphi^{\lambda} + \beta_1\varphi^n\right)/2$$

$$+ \delta_{zz}\left(\varphi^{n+1} + \beta_1\varphi^n\right)/2 \qquad \text{(A-62)}$$

Let $\varphi^{n+1} = \varphi^n = \varphi$, then the differential equation can be recognized as

$$\left(u\varphi_x\right)_x + \varphi_{yy} + \varphi_{zz} - \frac{4(1-\beta_1)}{(1+\beta_1)\,\epsilon\Delta t}\varphi_x + \frac{2(1-\beta_1)^2}{\epsilon\,\Delta t^2(1+\beta_1)}\varphi = 0 \qquad \text{(A-63)}$$

Letting $\Delta t \to 0$ yields the classic unsteady equation for the perturbation potential.

APPENDIX B

DERIVATION OF THE ADI DIFFERENCE EQUATIONS FOR SWEPT AND TAPERED WINGS

B.1 BASIC COORDINATE TRANSFORMATION

In analyzing swept wings, it is convenient to use a coordinate transformation in which the leading edge and trailing edge are defined by single values of the streamwise coordinate. The partial differential equation for time-dependent transonic small perturbation flow is given by

$$\left(2\phi_{xt} + \phi_{tt}\right)/\epsilon = \left(u\phi_x\right)_x + \phi_{yy} + \phi_{zz} \tag{B-1}$$

This equation is the classical form of the transonic small perturbation equation and does not include the higher order swept wing terms. (To obtain the basic steady state potential, φ_0 these higher order terms must be included to locate correctly the shock wave.)

A coordinate transformation, which meets the foregoing criteria, is shown in figure B-1 and is described by the following equation,

$$\begin{aligned} \xi &= \left[x - X_{LE}(y)\right]/S(y) - 1 \\ \eta &= y \\ \zeta &= z \end{aligned} \tag{B-2}$$

where S(y) is the semichord of the wing cross section at the spanwise location y. Then

$$\begin{aligned} \phi_x &= \phi_\xi/S \qquad \phi_y = \left(\phi_\xi \xi_y + \phi_\eta\right) \\ \phi_z &= \phi_\zeta \end{aligned} \tag{B-3}$$

The differential equation (B-1) becomes

$$\left(2\phi_{\xi t}/S + \phi_{tt}\right)/\epsilon = \frac{1}{S}\left(u\phi_\xi/S\right)_\xi + \xi_y \frac{\partial}{\partial \xi}\left(\phi_\xi \xi_y + \phi_\eta\right) + \frac{\partial}{\partial \eta}\left(\phi_\xi \xi_y + \phi_\eta\right) + \phi_{\zeta\zeta} \tag{B-4}$$

Multiplying by S yields

$$\left(2\phi_{\xi t} + S\phi_{tt}\right)/\epsilon = \left(u\phi_\xi/S\right)_\xi + \xi_y\left[S\left(\phi_\xi \xi_y + \phi_\eta\right)\right]_\xi + S\frac{\partial}{\partial \eta}\left(\phi_\xi \xi_y + \phi_\eta\right) + S\phi_{\zeta\zeta} \tag{B-5}$$

The second and third terms from the last can be combined to obtain a conservation form of the differential equation. Thus

$$\left(2\phi_{\xi t} + S\phi_{tt}\right)/\epsilon = \left(u\phi_\xi/S\right)_\xi + \left[S\xi_y\left(\phi_\xi \xi_y + \phi_\eta\right)\right]_\xi + \left[S\left(\phi_\xi \xi_y + \phi_\eta\right)\right]_\eta + (S\phi)_{\zeta\zeta} \tag{B-6}$$

For convenience, we define the following quantities

$$X = S\xi_y(\phi_\xi \xi_y + \phi_\eta) \tag{B-7}$$

$$Y = S\phi_\xi \xi_y \tag{B-8}$$

Then the differential equation becomes

$$(2\phi_{\xi t} + S\phi_{tt})/\epsilon = (u\phi_\xi\phi_\xi)_\xi + X_\xi + Y_\eta + \frac{\partial}{\partial\eta}(S\phi_\eta) + \frac{\partial}{\partial\zeta}(S\phi_\zeta) \tag{B-9}$$

Following Borland, Rizzetta, and Yoshihara (ref. 17) we construct an ADI procedure for the solution of this differential equation. We obtain

For the ξ sweep:

$$(2/\epsilon)\overleftarrow{\delta}_\xi\left(\frac{\phi^\alpha - \phi^n}{\Delta t}\right) = \delta_\xi[(u/S)(\phi^\alpha + \phi^n)/2] + \delta_\eta\left(S\delta_\eta\phi^n\right) + S\delta_{\zeta\zeta}\phi^n + \delta_\xi X^n + \delta_\eta Y^n \tag{B-10}$$

For the η sweep:

$$[(2/(\epsilon\,\Delta t)]\overleftarrow{\delta}_\xi(\phi^\lambda - \phi^\alpha) = \delta_\eta\left[S\left(\delta_\eta\phi^\lambda - \delta_\eta\phi^n\right)\right]/2 \tag{B-11}$$

For the ζ sweep:

$$[(2/(\epsilon\,\Delta t^2)]\,(\phi^{n+1} - 2\phi^n + \phi^{n-1}) + [2/(\epsilon\,\Delta t)]\overleftarrow{\delta}_\xi(\phi^{n+1} - \phi^\lambda) = \frac{S}{2}\delta_{\zeta\zeta}(\phi^{n+1} - \phi^n) \tag{B-12}$$

As in the rectangular wing, we represent the nth approximation and the intermediate approximations between the nth and n + 1st in the form

$$\phi^n = e^{i\,n\,\omega\,\Delta t}\varphi^n, \quad \phi^\lambda = e^{-i(n+1)\,\omega\,\Delta t}\varphi^\lambda$$

$$\phi^\alpha = e^{i(n+1)\,\omega\,\Delta t}\varphi^\alpha$$

Substituting into equations (B-10), (B-11), and (B-12), we obtain the following equations:

For the ξ sweep:

$$[2/(\epsilon\,\Delta t)]\overleftarrow{\delta}_\xi\left(\varphi^\alpha - \beta_1\varphi^n\right) = \delta_\xi\left[(u/S)\delta_\xi\left(\varphi^\alpha + \beta_1\varphi^n\right)\right]/2 + \beta_1\delta_\eta\left(S\delta_\eta\varphi^n\right) + \beta_1\delta_\zeta\left(S\delta_\zeta\varphi^n\right) + \beta_1\delta_\xi X^n + \beta_1\delta_\eta Y^n \tag{B-13}$$

where $X^n = S\xi_\eta(\delta_\xi\varphi^n\xi_\eta + \delta_\eta\varphi^n)$ and $Y^n = S\delta_\xi(\varphi^n\xi_\eta)$. Writing the equation with φ^α on the left hand side yields

$$[2/(\epsilon\,\Delta t)]\overleftarrow{\delta}_\xi\varphi^\alpha - \delta_\xi\Big[(u/S)\,\delta_\xi\varphi^\alpha\Big]/2 = \beta_1\Big\{[2/(\epsilon\,\Delta t)]\overleftarrow{\delta}_\xi\varphi^n$$

$$+ \delta_\xi\Big[(u/S)\,\delta_\xi\varphi^n\Big]/2 + \delta_\eta\Big(S\delta_\eta\varphi^n\Big) + S\delta_{\zeta\zeta}\varphi^n + \delta_\xi X^n + \delta_\eta Y^n\Big\} \qquad \text{(B-14)}$$

For the η sweep:

$$[2/(\epsilon\,\Delta t)]\overleftarrow{\delta}_\xi\Big(\varphi^\lambda - \varphi^\alpha\Big) = \delta_\eta\Big[S\Big(\delta_\eta\varphi^\lambda - \beta_1\delta_\eta\varphi^n\Big)\Big]/2$$

Writing the equation with φ^λ on the left hand side yields

$$[2/(\epsilon\,\Delta t)]\overleftarrow{\delta}_\xi\varphi^\lambda - \delta_\eta\Big(S\delta_\eta\varphi^\lambda\Big)/2 = [2/(\epsilon\,\Delta t)]\overleftarrow{\delta}_\xi\varphi^\alpha - \beta_1\Big(S\delta_\eta\varphi^n\Big)/2 \qquad \text{(B-15)}$$

For the ζ sweep:

$$[S/(\epsilon\,\Delta t)]\Big(\varphi^{n+1} - 2\beta_1\varphi^n + \beta_1^2\varphi^{n-1}\Big) + [2/(\epsilon\,\Delta t)]\overleftarrow{\delta}_\xi\Big(\varphi^{n+1} - \varphi^\lambda\Big) = \frac{S}{2}\,\delta_{\zeta\zeta}\Big(\varphi^{n+1} - \beta_1\varphi^n\Big)$$

writing the equation with φ^{n+1} on the left hand side yields, after dividing by S,

$$\delta_{\zeta\zeta}\varphi^{n+1}/2 - \varphi^{n+1}/\big(\epsilon\,\Delta t^2\big) - [2/(\epsilon\,\Delta t\,S)]\overleftarrow{\delta}_\xi\varphi^{n+1}$$

$$= \beta_1\Big[\delta_{\zeta\zeta}\varphi^n/2 - \Big[1/\big(\epsilon\,\Delta t^2\big)\Big]\Big(2\varphi^n - \beta_1\varphi^{n-1}\Big)\Big] - [2S/(\epsilon\,\Delta t)]\overleftarrow{\delta}_\xi\varphi^\lambda \qquad \text{(B-16)}$$

In the ξ sweep, when we replace u by $\bar{u} = u/S$, then the terms $\delta_\xi\,(\bar{u}\delta_\xi)$ are the same form as the unswept coordinate system. Similarly,

$$\delta_\eta\Big(S\delta_\eta\varphi^n\Big)/2 = a_j S_{j-\,1/2}\Big(\varphi^n_{i\,j-1\,k} - \varphi^n_{ijk}\Big) - b_j S_{j+\,1/2}\Big(\varphi^n_{ijk} - \varphi^n_{i\,j+1\,k}\Big) \qquad \text{(B-17)}$$

In the ξ and η sweeps, replacing

$$a_j S_{j-\,1/2} = \bar{a}_j$$

$$b_j S_{j+\,1/2} = \bar{b}_j \qquad \text{(B-18)}$$

the term $\delta_\eta(S\delta_\eta\varphi^n)$ becomes

$$2\,\bar{a}_j\Big(\varphi^n_{i\,j-1\,k} - \varphi^n_{ijk}\Big) - 2\,\bar{b}_j\Big(\varphi^n_{ijk} - \varphi^n_{i\,j+1\,k}\Big) \qquad \text{(B-19)}$$

The difference equations for the ξ sweep become

$$c_{3i}\left(\varphi^{\alpha}_{ijk} - \varphi^{\alpha}_{i-1\,jk}\right) - c_i\bar{u}_{i+1\,j}\left(\varphi^{\alpha}_{i+1\,jk} - \varphi^{\alpha}_{ijk}\right) + d_i\bar{u}_{ij}\left(\varphi^{\alpha}_{ijk} - \varphi^{\alpha}_{i-1\,jk}\right)$$

$$= \beta_1\Big[c_{3i}\left(\varphi^{n}_{ijk} - \varphi^{n}_{i-1\,jk}\right) + c_i\bar{u}_{i+1\,j}\left(\varphi^{n}_{i+1\,jk} - \varphi^{n}_{ijk}\right)$$

$$- d_i\bar{u}_{ij}\left(\varphi^{n}_{ijk} - \varphi^{n}_{i-1\,jk}\right) + 2\bar{a}_j\left(\varphi^{n}_{i\,j-1\,k} - \varphi^{n}_{ijk}\right)$$

$$- 2\bar{b}_j\left(\varphi^{n}_{ijk} - \varphi^{n}_{i\,j+1\,k}\right) + 2a_kS_j\left(\varphi^{n}_{ij\,k-1} - \varphi^{n}_{ijk}\right)$$

$$- 2b_k\left(\varphi^{n}_{ijk} - \varphi^{n}_{ij\,k+1}\right) + \delta_\xi X^n + \delta_\eta Y^n\Big] \qquad \text{(B-20)}$$

where $c_{3i} = 2/[\varepsilon\,\Delta t(X_i - X_{i-1})]$. The difference equations for the ξ sweep of the swept wing differ in form from the rectangular version only in the factor S_j multiplying a_k and b_k and the addition of the cross terms, $\delta_\xi X^n - \delta_\eta Y^n$.

We require the difference form of the derivatives, $\delta_\xi X^n + \delta_\eta Y^n$, for the right hand side of the equations. With $G(\xi,\eta) = \xi_y$ we have

$$\delta_\xi X^n = \delta_\xi\Big[SG\left(G\delta_\xi\varphi^n + \delta_\eta\varphi^n\right)\Big] = \delta_\xi\left(SG^2\delta_\xi\varphi^n\right) + \delta_\xi\left(G\delta_\eta\varphi^n\right)S \qquad \text{(B-21)}$$

The first term is the same form as $\delta_\xi(\bar{u}\delta_\xi\varphi^n)$, with G^2 replacing u, and can be easily written down. The Y^n term is

$$\delta_\eta Y^n = \delta_\eta\left(SG\delta_\xi\varphi^n\right) \qquad \text{(B-22)}$$

From reference 1, the second order difference for δ_ξ is seen to be given by

$$\delta_\xi\varphi\,\Big|_{ijk} = c_{1i}\left(\varphi_{i+1\,jk} - \varphi_{ijk}\right) + d_{1i}\left(\varphi_{ijk} - \varphi_{i-1\,jk}\right) \qquad \text{(B-23)}$$

where c_{1i} and d_{1i} are defined on page 40 of reference 1. Similarly, we write

$$\delta_\eta\varphi = c_{1j}\left(\varphi_{i\,j+1\,k} - \varphi_{ijk}\right) + d_{1j}\left(\varphi_{ijk} - \varphi_{1\,j-1\,k}\right) \qquad \text{(B-24)}$$

where the subscript j denotes that c_{1j} is defined in the same way as c_{1i} but with the variables y_j.
Combining the two difference equations, we obtain for the cross derivative term

$$\delta_\xi\left[G\delta_\eta\varphi\right] = c_{1i}G_{i+1\,j}H_{i+1\,jk} - \left(c_{1j} - d_{1i}\right)G_{ij}H_{ijk} - d_{1i}G_{i-1\,j}H_{i-1\,jk} \qquad \text{(B-25)}$$

where $H_{ijk} = c_{ij}\left(\varphi_{ij+1k} - \varphi_{ijk}\right) + d_{1j}\left(\varphi_{ijk} - \varphi_{ij-1k}\right)$
Similar results may be written down for $\delta_\eta[SG\delta_\xi\varphi]$. No special application of boundary conditions is required since the boundary values of φ are calculated from the boundary conditions at each sweep.

Similarly, we obtain for the η sweep

$$\bar{a}_j\left(\varphi^{\lambda}_{i\,j-1\,k} - \varphi^{\lambda}_{ijk}\right) - \bar{b}_j\left(\varphi^{\lambda}_{ijk} - \varphi^{\lambda}_{i\,j+1\,k}\right)$$
$$- c_{3i}\left(\varphi^{\lambda}_{ijk} - \varphi^{\lambda}_{i-1\,jk}\right) = - c_{3i}\left(\varphi^{\alpha}_{ijk} - \varphi^{\alpha}_{i-1\,jk}\right)$$
$$+ \beta_1\left[\bar{a}_j\left(\varphi^{n}_{i\,j-1\,k} - \varphi^{n}_{ijk}\right) - \bar{b}_j\left(\varphi^{n}_{ijk} - \varphi^{n}_{i\,j+1\,k}\right)\right] \tag{B-26}$$

This is seen to have the same form as the rectangular wing version.

Finally, the ζ sweep becomes

$$a_k\left(\varphi^{n+1}_{ij\,k-1} - \varphi^{n+1}_{ijk}\right) - b_k\left(\varphi^{n+1}_{ijk} - \varphi^{n}_{ij\,k+1}\right) - E_1\varphi^{n+1}_{ijk}$$
$$- S_j c_{3i}\left(\varphi^{n+1}_{ijk} - \varphi^{n+1}_{i-1\,jk}\right)$$
$$= \beta_1\left[a_k\left(\varphi^{n}_{ij\,k-1} - \varphi_{ijk}\right) - b_k\left(\varphi^{n}_{ijk} - \varphi^{n}_{ij\,k+1}\right)\right.$$
$$\left. - E_1\left(\varphi^{n}_{ijk} - \beta_1\varphi^{n+1}_{ijk}\right)\right] - S_j c_{3i}\left(\varphi^{\lambda}_{ijk} - \varphi^{\lambda}_{i-1\,jk}\right) \tag{B-27}$$

This has the same form as the rectangular version except c_{3i} is replaced by $S_j c_{3i}$.

B.2 WING AND WAKE BOUNDARY CONDITIONS

The wing boundary conditions are of the general form

$$\varphi_z = f'(x) + i\,\omega\,f(x)$$

where $z = \delta f(x)e^{i\omega t}$ is the motion of each cross section. In terms of the variable ξ, this becomes

$$\varphi_z = f'(\xi)\xi_x + i\,\omega\,f(\xi)$$

Since

$$\xi = \left(x - X_{LE}\right)/S - 1$$

then

$$\varphi_z = f'(\xi)/S + i\,\omega\,f(\xi) \tag{B-28}$$

This differs from the rectangular version in the term S_j dividing the slope $f'(\xi)$. For the boundary conditions on the wake, we have

$$\Delta\varphi_x + i\,\omega\,\Delta\varphi = 0$$

or

$$\Delta\varphi_\xi + i\,\omega\,S\,\Delta\varphi = 0 \tag{B-29}$$

In the wake boundary conditions, the reduced frequency is replaced by the product of reduced frequency and S_j.

B.3 MESH BOUNDARY CONDITIONS

On the upstream boundary, the condition for outgoing plane waves is

$$\phi_x - M\phi_t/(1-M) = 0$$

In swept wing coordinates this becomes

$$\phi_\xi - MS\phi_t/(1-M) = 0 \tag{B-30}$$

This is the same form as the rectangular wing version except for the factor of the semichord on the quantity M/(1-M). Hence the parameter C_{k1} now depends upon j. We then have for the boundary conditions

$$\varphi^{n+1}_{1jk} = \overline{C}_{k1j}\varphi^{n+1}_{2jk} + \beta_1\left(\varphi^{n}_{2jk} - \overline{C}_{k1j}\varphi^{n}_{1jk}\right) \tag{B-31}$$

where

$$C_{k1j} = (\xi_2 - \xi_1)\, MS_j/(1-M)\Delta t$$

$$\overline{C}_{k1j} = \left(1 - C_{k1j}\right)/\left(1 + C_{k1j}\right) \tag{B-32}$$

Similarly, for the downstream boundary

$$\phi_\xi + MS\phi_t / (1+M) = 0$$

This leads to

$$\varphi^{n+1}_{i_{max}\,jk} = \overline{C}_{k3j}\varphi^{n+1}_{i_{max}-1\,jk} + \beta_1\left(\varphi^{n}_{i_{max}-1\,jk} - \overline{C}_{k3j}\,\varphi^{n}_{i_{max}\,jk}\right) \tag{B-33}$$

where

$$\overline{C}_{k3j} = \left(1 - C_{k3j}\right)/\left(1 + C_{k3j}\right)$$

$$C_{k3j} = MS_j\left(\xi_{i_{max}} - \xi_{i_{max}-1}\right)/(1+M)\,\Delta t \tag{B-34}$$

B.4 CONDITIONS OF SYMMETRY AT THE ROOT CHORD PLANE

At the root chord plane $y = \eta = 0$, we apply the condition of symmetry given by

$$\varphi_y = 0$$

In the swept wing coordinate system this becomes

$$\varphi_\xi \xi_y + \varphi_\eta = 0$$

We let $G(\xi,\eta) = \xi_y$ and let $j = 2$ be the plane of symmetry $y = 0$. In difference form, the conditions of symmetry at the point x_i become

$$G_{i2}\left(\varphi^{\lambda}_{i2j} - \varphi^{\lambda}_{i-12j}\right)/(\xi_i - \xi_{i-1}) + \left(\varphi^{\lambda}_{i3k} - \varphi^{\lambda}_{i1k}\right)/(\eta_3 - \eta_1) = 0 \tag{B-35}$$

Here we use backward difference in ξ since we sweep in the direction of increasing i. Since $\eta_1 = -\eta_3$, we define

$$c_{4i} = G_{i2}(\eta_3 - \eta_1)/(\xi_i - \xi_{i-1}) = 2\eta_3 G_{i2}/(\xi_i - \xi_{i-1}) \tag{B-36}$$

The boundary condition then becomes

$$c_{4i}\left(\varphi^{\lambda}_{i2k} - \varphi^{\lambda}_{i-2\,k}\right) + \varphi^{\lambda}_{i3k} - \varphi^{\lambda}_{i1k} = 0$$

from which

$$\varphi^{\lambda}_{i1k} = \varphi^{\lambda}_{i3k} - c_{4i}\left(\varphi^{\lambda}_{i2k} - \varphi^{\lambda}_{i-1\,2\,k}\right) \tag{B-37}$$

For the outer boundary condition, we have

$$\phi_y + \sqrt{K}\ M\phi_t/(1-M^2) = 0$$

In swept wing coordinate, this becomes

$$\phi_\xi \xi_y + \phi_\eta + \sqrt{K}M\phi_t/(1-M^2) = 0 \tag{B-38}$$

We write equation (B-38) in implicit difference form, for $j = j_{max} - 1$. We obtain

$$G_{ij}\left(\phi^{n+1/2}_{ijk} - \phi^{n+1/2}_{i-1\,jk}\right)/(\xi_i - \xi_{i-1}) + \frac{\phi^{n+1/2}_{i\,j+1\,k} - \phi^{n+1/2}_{ijk}}{\eta_{j+1} - \eta_j}$$

$$+ \frac{\sqrt{K}\,M}{(1-M^2)\Delta t}\left(\phi^{n+1}_{i\,j+1/2\,k} - \phi^{n}_{i\,j+1/2\,k}\right) = 0 \tag{B-39}$$

Let $c_{2i} = G_{i\,j_{max}-1}(\eta_{j_{max}} - \eta_{j_{max}-1})/(\xi_i - \xi_{i-1})$

and $C_{k2} = M\sqrt{K}(\eta_{j_{max}} - \eta_{j_{max}-1})/[(1-M^2)\Delta t]$

then the boundary conditions become

$$c_{2i}\left(\varphi^{n+1}_{ijk} + \beta_1\varphi^{n}_{ijk} - \varphi^{n+1}_{i-1\,jk} - \beta_1\varphi^{n}_{i-1\,jk}\right) + \varphi^{n+1}_{i\,j+1\,k} + \beta_1\varphi^{n}_{i\,j+1\,k}$$

$$- \varphi^{n+1}_{ijk} - \beta_1\varphi^{n}_{ijk} + C_{k2}\left(\varphi^{n+1}_{i\,j+1\,k} + \varphi^{n+1}_{ijk} - \beta_1\varphi^{n}_{i\,j+1\,k} - \beta_1\varphi^{n}_{ijk}\right) = 0 \tag{B-40}$$

Rearranging the terms leads to

$$\varphi^{n+1}_{i\,j_{max}\,k} = (\overline{C}_{k2} - \bar{c}_{2i})\varphi^{n+1}_{i\,j_{max}-1\,k} + \bar{c}_{2i}\varphi^{n+1}_{i-1\,j_{max}-1\,k}$$

$$- \beta_1\left[\bar{c}_{2i}\left(\varphi^{n}_{i\,j_{max}-1\,k} - \varphi^{n}_{i\,j_{max}-2\,k}\right) + \overline{C}_{k2}\varphi^{n}_{ij\,max\,k} - \varphi^{n}_{i\,j_{max}-1\,k}\right] \tag{B-41}$$

where $\bar{c}_{2i} = c_{2i}/(1 + C_{k2})$ and $\overline{C}_{k2} = (1 - C_{k2})/(1 + C_{k2})$.

The boundary conditions on the wing and the wake are treated in the same way as for the rectangular wing, with the additional factor of the semichord as described in the preceding section. The boundary conditions on the outer ζ mesh boundaries are unchanged from the rectangular wing since the ζ variable is unaffected by the swept wing transformation.

B.5 DERIVATION OF THE SWEPT WING TRANSFORMATION

To illustrate the method of deriving the swept wing coordinate transformation, we consider the simple swept tapered wing whose leading and trailing edges are straight lines. Let θ_{sw} be the swept angle of the leading edge and R be the taper ratio. Then the x coordinate of the leading edge is given by

$$X_{LE} = -1 + y \tan \theta_{sw} \tag{B-42}$$

For the trailing edge, we write

$$X_{TE} = 1 + by \tag{B-43}$$

and determine b so that at $y = y_t$ (that is, at the tip), the chord of the wing is equal to 2R. We then obtain

$$b = 2(R-1)/y_t + \tan \theta_{sw} = 2C_2 + C_1 \tag{B-44}$$

where

$$C_2 = (R-1)/y_t \text{ and } C_1 = \tan \theta_{sw}. \tag{B-45}$$

Because of the slope singularity in the pressure at the wing tip, the tip is placed halfway between grid points in the spanwise direction. The grid lines which coincide with the leading and trailing edges of the wing are extended beyond the last spanwise grid point on the wing planform, j = js, using a quadratic to preserve continuity of slope. At a point midway between the wing tip and the intersection of the linear extensions of the

edges, y_m, the slopes of the edge coordinate lines are both set equal to the average of the edge slopes at the tip. The edge coordinate lines are extended beyond the point y_m in linear fashion. The point of intersection of the edge extensions is found by setting $x_{LE} = x_{TE}$ in equations (B-42) and (B-43).

$$y = -1/2C_2 \tag{B-46}$$

Let y_{jm} be the grid value of y for which

$$y_{jm} < -1/4C_2 < y_{jm+1} \tag{B-47}$$

Thus for $j < jm$, we define

$$X_{LE} = -1 + C_{1y} + b_1(y - y_{js})^2$$

$$X_{TE} = 1 + (C_1 + 2C_2)y + b_2(y - y_{js})^2 \tag{B-48}$$

The coefficients b_1 and b_2 are determined so that at $y = y_{jm}$,

$$X'_{LE} = X'_{TE} = [C_1 + (C_1 + 2C_2)]/2 = C_1 + C_2$$

This leads to two equations for b_1 and b_2 which yield

$$b_2 = C_2/2(y_{jm} - y_{js}) = -b_1$$

Beyond $y = y_{jm}$, the trailing edge and leading edge are given by

$$X_{LE} = -1 + C_1 y_{jm} + b_1(y_{jm} - y_{js})^2 + (C_1 + C_2)(y - y_{jm})$$

$$X_{TE} = 1 + (C_1 + 2C_2)y_{jm} + b_2(y_{jm} - y_{js})^2 + (C_1 + C_2)(y - y_{jm})$$

In summary, we have, for $0 \le y \le y_{js}$,

$$X_{LE} = -1 + C_1 y \tag{B-49}$$

$$X_{TE} = 1 + (C_1 + 2C_2)y \tag{B-50}$$

$$S(y) = (X_{TE} - X_{LE})/2 = 1 + C_2 y \tag{B-51}$$

for $y_{js} \le y \le y_{jm}$,

$$X_{LE} = -1 + C_1 y - C_2(y - y_{js})^2/2(y_{jm} - y_{js}) \tag{B-52}$$

$$X_{TE} = 1 + (C_1 + 2C_2)y + C_2(y - y_{js})^2/2(y_{jm} - y_{js}) \tag{B-53}$$

$$S(y) = 1 + C_2 y + C_2(y - y_{js})^2/2(y_{jm} - y_{js}) \tag{B-54}$$

and for $y > y_{jm}$,

$$X_{LE} = -1 + C_1 y_{jm} - C_2(y_{jm} - y_{js})/2 + (C_1 + C_2)(y - y_{jm}) \quad \text{(B-55)}$$

$$X_{TE} = 1 + (C_1 + 2C_2)y_{jm} + C_2(y_{jm} - y_{js})/2 + (C_1 + C_2)(y - y_{jm}) \quad \text{(B-56)}$$

$$S(y) = 1 + C_2 y_{jm} + C_2(y_{jm} - y_{js})/2 \quad \text{(B-57)}$$

The transformation is given by

$$\xi(\eta) = [x - X_{LE}(\eta)]/S(\eta) - 1 \quad \text{(B-58)}$$

The mapping function required at every grid point is

$$G(\xi,\eta) = \xi_\eta = - S'(\eta)(\xi + 1) - X'_{LE}/S(\eta) \quad \text{(B-59)}$$

where X'_{LE} and S' are found by differentiating the expressions in equations (B-49) through (B-57)).

B.6 ELIMINATION OF SLOPE DISCONTINUITY AT PLANE OF SYMMETRY

Some discontinuity affects at the line of symmetry can be alleviated by making the lead and trailing edges bend normal to the plane of symmetry. For $j \geq 4$, the coordinates of the leading and trailing edges are described in the foregoing section. The portion of the leading edge for $j = 2$ to $j = 4$ is fitted to a quadratic of the form

$$X_{LE} = X_{LE}(0) + ay^2 \quad \text{(B-60)}$$

The quantity a is determined to match the slope at $j = 4$. Hence,

$$a = C_1/2y_4$$

Thus at $j = 2$ or 3 we have

$$X_{LE} = X_{LE}(0) + C_1 y^2/2y_4$$

We determine $x_{LE}(0)$ so that at $y = y_4$, X_{LE} matches the known value, X_{LE4}. Thus

$$X_{LE}(0) + C_1 y_4/2 = X_{LE4}$$

and we finally obtain

$$X_{LE} = X_{LE4} - C_1\left(y_4 - y^2/y_4\right)/2 \quad \text{(B-61)}$$

Similarly,

$$X_{TE} = X_{TE4} - (C_1 + 2C_2)\left(y_4 - y^2/y_4\right)/2 \quad \text{(B-62)}$$

We see that

$$S(y) = S(y_4) - C_2\left(y_4 - y^2/y_4\right)/2 \qquad \text{(B-63)}$$

Equations (B-61) through (B-63) replace equations (B-49) through (B-51) for $j = 2$ and 3. For larger sweep angles, the region should contain more than $j = 2$ to $j = 4$ to adequately represent the geometry in the differencing.

APPENDIX C
REVISED DIFFERENCE OPERATORS

C.1 THE δ_{tt} OPERATOR FOR A CHANGE IN TIME STEP

The differencing in the equation for the x and y sweeps remains unchanged when the time step Δt is changed. In the z sweep, the only term which is affected is the second derivative with respect to time. Let the time step between the n-1 and the nth approximation be Δt_0 and between the n and n + 1st approximation be Δt_1. We express the time derivative in difference form as

$$a\phi^{n+1} + b\phi^{n} + c\phi^{n-1} = \phi_{tt}$$

and expand the equation about the nth approximation. We then determine a, b, and c so that the left hand side gives ϕ_{tt} at the nth approximation. Thus we obtain

$$a\left[\phi + \Delta t_1\phi_t + \frac{\Delta t_1^2}{2}\phi_{tt} + \cdots\right] + b\phi + c\left[\phi - \Delta t_0\phi_t + \frac{\Delta t_0^2}{2}\phi_{tt} + \cdots\right] = \phi_{tt}$$

Equating coefficients on the right and left sides of the equation yields

$$a + b + c = 0$$

$$a\Delta t_1 - c\Delta t_0 = 0$$

$$a\Delta t_1^2 + c\Delta t_0^2 = 2$$

Solving the last two equations simultaneously yields

$$a = 2/\left[\Delta t_1(\Delta t_0 + \Delta t_1)\right]$$

$$c = 2/\left[\Delta t_0(\Delta t_0 + \Delta t_1)\right]$$

from which we obtain

$$b = -\frac{2}{(\Delta t_0 + \Delta t_1)}\left[\frac{1}{\Delta t_1} + \frac{1}{\Delta t_0}\right]$$

Introducing the variable $R = \Delta t_1/\Delta t_0$ and $E_0 = (1+R)/2R$, $E_1 = 1/[\varepsilon E_0 \Delta t_1^2]$; then

$$\delta_{tt}\phi/\epsilon = E_1\left[\phi^{n+1} - 2E_0R\phi^{n} + R\phi^{n-1}\right]$$

C.2 SECOND ORDER BACKWARD DIFFERENCE FOR δ_x

For the first derivative with respect to x at the point x_i, we write

$$a(\varphi_{ij} - \varphi_{i-1j}) - b(\varphi_{i-1j} - \varphi_{i-2j}) = \varphi_x$$

Expanding about the point x_i, we get

$$a\left[\varphi - \varphi + \Delta x_1 \varphi_x - \frac{\Delta x_1^2}{2} \varphi_{xx} + \cdots\right]$$

$$- b\left[\varphi - \Delta x_1 \varphi_x + \frac{\Delta x_1^2}{2} \varphi_{xx} + \cdots - \varphi + (\Delta x_1 + \Delta x_2)\varphi_x - \frac{(\Delta x_1 + \Delta x_2)^2}{2} \varphi_{xx} + \cdots\right] = \varphi_x$$

where $\Delta x_1 = x_i - x_{i-1}$ and $\Delta x_2 = x_{i-1} - x_{i-2}$. Setting the coefficient of φ_x on the left hand side equal to unity and the coefficient of φ_{xx} equal to zero yields

$$a\,\Delta x_1 - b\,\Delta x_2 = 1$$

$$a\,\Delta x_1^2 + b\left[\Delta x_1^2 - (\Delta x_1 + \Delta x_2)^2\right] = 0$$

We solve the two equations simultaneously for a and b and obtain

$$a = (2\,\Delta x_1 + \Delta x_2)/[\Delta x_1(\Delta x_1 + \Delta x_2)]$$

$$b = \Delta x_1/\Delta x_2\,(\Delta x_1 + \Delta x_2)$$

Writing

$$c_{3i}(\varphi_{ij} - \varphi_{i-1\,j}) - d_{3i}(\varphi_{i-1\,j} - \varphi_{i-2\,j}) \simeq \delta_x$$

yields finally

$$c_{3i} = [2(x_i - x_{i-1}) + x_{i-1} - x_{i-2}]/[(x_i - x_{i-2})(x_i - x_{i-1})]$$

$$d_{3i} = (x_i - x_{i-1})/[(x_i - x_{i-2})(x_{i-1} - x_{i-2})]$$

APPENDIX D
SHOCK BOUNDARY CONDITIONS FOR THE ADI METHOD

Shock boundary conditions were derived by Hafez, Riszk, and Murman (ref. 37), by Williams (ref. 38), and by Seebass, Yu, and Fung (ref. 39). The results were reported in reference 6 and extended to harmonic motion for application to the direct solution of the difference equations. The condition that the potential be continuous across the shock is given by

$$[\varphi] = 0$$

where [ ] denotes the jump in the quantity across the shock. The condition of continuity of mass across the shock from equation (A-11) of reference 7 is

$$\frac{2}{\epsilon}\frac{\partial X}{\partial t} + \langle k - (\gamma + 1)\rangle = 0 \tag{D-1}$$

where $\langle\ \rangle$ denotes mean value of the quantity at the shock. We now expand the shock conditions about the steady location X_0, with the shock position at time t being given by $X_0 + aX_1$. Then continuity of the potential across the shock becomes

$$[\varphi] = [\varphi_0 + a\varphi_1] + aX_1[\varphi_{0x} + a\varphi_{1x} + \cdots]$$

$$[\varphi_0] = 0 \tag{D-2}$$

$$[\varphi_1] + X_1[\varphi_{0x}] = 0 \tag{D-3}$$

Similarly, expanding equation (D-1) yields

$$\frac{2a}{\epsilon}\frac{\partial X_1}{\partial t} + \langle k - (\delta + 1)[\varphi_{0x} + a\varphi_{1x} + aX_1(\varphi_{0xx} + a\varphi_{1xx}) + \cdots]\rangle = 0 \tag{D-4}$$

Equating the coefficients of each power of a equal to zero yields

$$\langle k - (\gamma+1)\varphi_{0x}\rangle = \langle u\rangle = 0 \tag{D-5}$$

$$\frac{2}{\epsilon}\frac{\partial X_1}{\partial t} - (\delta + 1)\langle \varphi_{1x} + X_1\varphi_{0xx}\rangle = 0 \tag{D-6}$$

Eliminating X_1 by equation (D-3) yields

$$\frac{2}{\epsilon}\frac{\partial[\varphi_1]}{\partial t} + (\gamma+1)[\varphi_{0x}]\langle \varphi_{1x}\rangle - (\gamma+1)\langle \varphi_{0xx}\rangle[\varphi_1] = 0 \tag{D-7}$$

Since $K - (\gamma+1)\varphi_{0x} = u$, we have

$$\frac{2}{\epsilon}\frac{\partial[\varphi_1]}{\partial t} - [u]\langle\varphi_{1x}\rangle + \langle u_x\rangle[\varphi_1] = 0 \tag{D-8}$$

In implicit difference form, this becomes

$$\frac{2}{\epsilon\,\Delta t}[\tilde{\varphi}_1] + \langle u_x \rangle [\tilde{\varphi}_1]/2 - [u]\langle \tilde{\varphi}_{1x} \rangle/2$$

$$= \beta_1 \left\{ \frac{2}{\epsilon\,\Delta t}\left[\varphi_1^n\right] - \langle u_x \rangle \left[\varphi_1^n\right]/2 + [u]\langle \varphi_{1x}^n \rangle/2 \right\} \tag{D-9}$$

In terms of the variables on the grid, we write

$$C_{03} = [u] = u_{i+1\,j} - u_{ij}$$

$$C_{01} = 2/(\epsilon\,\Delta t)$$

$$C_{02} = \langle u_x \rangle \tag{D-10}$$

From the equation following (A-22) of reference 7, we have

$$C_{02} = \frac{u_{ij} - u_{i-1j}}{x_i - x_{i-2}} + \frac{u_{i+2\,j} - u_{i+1j}}{x_{i+2} - x_i} \tag{D-11}$$

Then the equation becomes

$$(C_{01} + C_{02}/2)[\tilde{\varphi}_1] - C_{03}\langle \tilde{\varphi}_{1x} \rangle/2 = \beta_1 \left\{ (C_{01} - C_{02}/2)\left[\varphi_1^n\right] + C_{03}\langle \varphi_{1x}^n \rangle/2 \right\} \tag{D-12}$$

We now express the jump and average values across the shock in terms of the values of φ at the grid points for the point x_i for which $u_{i+1\,j} > 0$ and $u_{ij} < 0$. We have

$$[\tilde{\varphi}_1] = \tilde{\varphi}_{ij} - \tilde{\varphi}_{i-1\,j} \tag{D-13}$$

$$\langle \tilde{\varphi}_{1x} \rangle = \frac{1}{2}\left\{ \frac{\tilde{\varphi}_{i+1\,j} - \tilde{\varphi}_{ij}}{x_{i+1} - x_i} + \frac{\tilde{\varphi}_{i-1\,j} - \tilde{\varphi}_{i-2\,j}}{x_{i-1} - x_{i-2}} \right\} \tag{D-14}$$

We let $C_{04} = C_{03}/4(x_{i+1} - x_i)$ and $C_{05} = C_{03}/4(x_{i-1} - x_{i-2})$. Then we get

$$(C_{01} + C_{02}/2)(\tilde{\varphi}_{ij} - \tilde{\varphi}_{i-1\,j}) - C_{04}(\tilde{\varphi}_{i+1\,j} - \tilde{\varphi}_{ij}) - C_{05}(\tilde{\varphi}_{i-1\,j} - \tilde{\varphi}_{i-2\,j})$$

$$= \beta_1 \left[(C_{01} - C_{02}/2)\left(\varphi_{ij}^n - \varphi_{i-1\,j}^n\right) + C_{04}\left(\varphi_{i+1\,j}^n - \varphi_{ij}^n\right) \right.$$

$$\left. + C_{05}\left(\varphi_{i-1\,j}^n - \varphi_{i-2\,j}^n\right) \right] \tag{D-15}$$

This equation replaces the difference equation in the x sweep.

APPENDIX E
A CONJUGATE GRADIENT ALGORITHM FOR ASYMMETRIC INDEFINITE COMPLEX MATRICES

In this section, we present a conjugate-gradient-type algorithm for unsymmetric complex matrices. The algorithm has an asymptotic convergence rate inversely proportional to the square root of the condition number of the coefficient matrix and does not assume the coefficient matrix to have any special property other than nonsingularity.

In section E.1, we shall present the principle of the classical conjugate gradient method. In section E.2, we shall present our algorithm and related theories.

E.1 DESCRIPTION OF THE CONJUGATE GRADIENT METHOD

Suppose we are interested in finding the solution x of the equation

$$A x = b \tag{E-1}$$

In the case that A is symmetric (or Hermitian in the case of complex matrices) and positive definite, the k-th step of the classical conjugate gradient method finds an approximate solution x_k for equation (E-1), so that the norm of the residual r_k is minimum in the space spanned by the vectors $\{Av_1, Av_2, \ldots, Av_k\}$, where $v_1, v_2, \ldots, v_k$ are orthonormal vectors, that is, $(v_i, v_j) = 0$, for $i \neq j$, and the lengths of the v_i's are equal to one. These orthonormal vectors are the backbone of the conjugate gradient method, and x_k is a linear combination of these v_j's.

Provided these v_j's can be computed easily, there is a whole class of algorithms which minimize the norm of the residual in the vector space spanned by $Av_1, Av_2, \ldots Av_k$ in the k-th step. These algorithms can be considered as variants of the classical conjugate gradient method. Among these are the conjugate residual, the modified conjugate residual, the variational method, SYMMLQ (Conjugate gradient method with LQ factorization for symmetric indefinite system), LSQR (Conjugate gradient method with QR factorization for least squares problems, A*A x = A*b, where A is rectangular matrix), etc. Chandra (ref. 33) gave a detailed comparison of the ones applicable to symmetric matrices.

For the sake of completeness, we present the classical conjugate gradient algorithm:

Algorithm 1. Classical Conjugate Gradient

Step 1: Choose x_0

Compute $r_0 = b - Ax_0$

Set $p_0 = r_0$

$a_0 = (r_0,r_0)/(p_0,Ap_0)$

$x_1 = x_0 + a_0p_0$

$r_1 = r_0 - a_0Ap_0$

Step 2: Compute

$a_i = (r_i,r_i)/(p_i,Ap_i)$,

$x_i + 1 = x_i + a_ip_i$,

$r_{i+1} = r_i - a_iAp_i$

$b_i = (r_{i+1}, r_{i+1})/(r_i,r_i)$

$p_{i+1} = r_{i+1} + b_ip_i$

Step 3: If r_{i+1} is small enough, terminate the iteration; else set $i = i + 1$ and go to step 2.

The vectors p_i are orthogonal to each other, although they are not unit vectors. The vectors v_i may be constructed by normalizing p_i.

In section E.1.1, we shall discuss orthogonalization. In section E.1.2, we shall discuss solution of equation (E-1) based on the orthonormal vectors.

E.1.1. ORTHOGONALIZATION

There are two classical orthogonalization processes for solutions of different numerical linear algebra problems.

First, there is the Lanczos' orthogonalization process for symmetric (or Hermitian) positive definite matrices. This process is computationally very efficient and extremely economical because it only requires the presence of v_k and v_{k-1} to compute v_{k+1}. If we let V_k be the matrix whose columns are $v_1, v_2, \ldots, v_k$, then the Lanczos' orthogonalization process at the k-th step has performed the following decomposition:

$$(V_k^*)AV_k = T_k \tag{E-2}$$

where T_k is a tridiagonal matrix.

More general for asymmetric matrices is the Gram-Schmidt orthogonalization process. This procedure does not have the computational efficiency of the Lanczos' process. To find the v_k vector, it requires presence of all the v_j's computed before. At the k-th step, the Gram-Schmidt orthogonalization process performs the following decomposition:

$$(V_k^*)AV_k = R_k \tag{E-3}$$

where R_k is an upper triangular matrix. When the dimension of A is large, we see that the Gram-Schmidt process is computationally infeasible.

The new algorithm we present here involves finding two sets of orthonormal vectors U and V. Similar to the Lanczos' process, it only requires the presence of v_{k-1} and v_k, and u_{k-1} and u_k to compute v_{k+1} and u_{k+1}. The k-th step of our algorithm performs the following decomposition:

$$U_k^*AV_k = T_k \tag{E-4}$$

E.1.2 COMPUTATION OF THE APPROXIMATE SOLUTION

Note that the second equation of Step 2 in Algorithm 1 computes the (i + 1)-th approximation of the solution. The vector p_i is sometimes called the directional vector. Other approaches proposed by Paige and Saunders (ref. 40), Bunch and Kaufman (ref. 41), and Chandra (ref. 33) to extend the classical conjugate gradient method to symmetric (Hermitian) nondefinite problems have been motivated by the following observation:

Let A be a symmetric (Hermitian) positive definite matrix. In the k-th step of Lanczos' orthogonalization process, the following matrix decomposition is performed:

$$V_k{}^*AV_k = T_k$$

If we choose $v_1 = r_0/\|r_0\|$, then x_k in equation (E-5) below is mathematically the same as the approximation to x in the k-th iteration of the classical conjugate gradient method:

$$T_k y_k = \|r_0\| e_1 \qquad \text{(E-5.a)}$$

and

$$x_k = x_0 + V_k y_k \qquad \text{(E-5.b)}$$

(e_1 is the vector with 1 in its first entry and zeroes everywhere else.)

Of course when A is symmetric (Hermitian) but not positive definite, the above statement is no longer true; but one can still solve equation (E-5) to obtain an approximate solution.

We have found that solving equation (E-5.a) by an LQ factorization (where L is lower triangular and Q is orthonormal) is numerically much more stable than generating directional vectors for the approximation of x_k's.

Thus there are two features in our new algorithm that are distinct from the classical conjugate gradient:

1. A new way of generating orthonormal vectors.
2. A tridiagonal system by an LQ factorization to obtain an approximation to the solution.

E.1.3. TRIDIAGONALIZATION ALGORITHM FOR UNSYMMETRIC MATRICES (USYMLQ)

As stated in section E.1.2, the k-th step, k = 1,2, ... of the algorithm to be described in this section involves the following decomposition:

$$U_k{}^*AV_k = T_k$$

where U_k and V_k are matrices whose columns are orthonormal and T_k is a tridiagonal matrix. We shall first describe our process with which we generate the orthonormal vectors, the u_k's and the v_k's:

Algorithm 2 (Tridiagonal Process)

(a) Set $u_0 = 0$,
$\beta_1 u_1 = b, \gamma_i v_1 = c$

(b)
(b.1) $p = Au_i - \gamma_i u_{i-1}$
(b.2) $q = A^*u_i - \beta_i v_{i-1}$
(b.3) $\alpha_i = u^*{}_i p \quad i = 1, 2, 3, \ldots$
(b.4) $\beta_{i+1} u_{i+1} = p - \alpha_i u_i$
(b.5) $\gamma_{i+1} v_{i+1} = q - \alpha^*{}_i v_i$

where b is the right hand side vector of the matrix equation we want to solve, and β_i and γ_i are chosen so that the vectors u_i and v_i will have norms equal to 1. We shall discuss the choice of c later in this section. For the moment, let us set c = b. The process terminates when β_i or γ_i equal zero.

In the rest of this section, we shall give an overall picture which relates the k-th approximate solution x_k to the k-th step of Algorithm 2. If we substitute (b.1) and (b.2) into equations (b.4) and (b.5) respectively and let U_k and V_k be nxk matrices whose columns are respectively the u_i's and the v_i's obtained from the first k steps of algorithm 2, and if we define a tridiagonal matrix T_k as

$$T_k = \begin{bmatrix} \alpha_1 & \gamma_2 & & & \\ \beta_2 & \alpha_2 & \gamma_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \ddots & \gamma_k \\ & & & \beta_k & \alpha_k \end{bmatrix}$$

where α_i, β_i, and γ_i are defined for each i in equation (b), then the first k steps of Algorithm 2 can be written in the following matrix equations:

$$AV_k = U_kT_k + \beta_{k+1}u_{k+1}e_k^T \qquad \text{(E-6.a)}$$

$$A^*U_k = V_kT_k^* + \gamma_{k+1}v_{k+1}e_k^T \qquad \text{(E-6.b)}$$

where e_k is a vector of length of n with 1 at the k-th entry and zero everywhere else. Multiplying (E-6.a) by an arbitrary k-vector, y_k, whose i-th element is η_i, we obtain

$$AV_ky_k = U_kT_ky_k + \beta_{k+1}u_ke_k^Ty_k$$

Since $b = U_k(\beta_1e_1)$ by definition, then if y_k and x_k are defined by the following equations

$$T_ky_k = \beta_1e_1 \qquad \text{(E-7.a)}$$

$$x_k = V_ky_k \qquad \text{(E-7.b)}$$

then we shall have

$$Ax_k = b + \eta_k\beta_{k+1}u_{k+1}$$

Hence x_k may be taken as an approximation to x, and will solve the original system if $\eta_k\beta_{k+1}$ is negligibly small. The above arguments are not complete, but they provide some motivation for defining the sequence of vectors x_k according to equations (E-7.a) and (E-7.b).

If x_k is as defined in equation (E-7), then the residual r_k will be

$$r_k = b - Ax_k \qquad \text{(E-8.a)}$$

$$= U_{k+1}(\beta_1 e_1 - S_ky_k) \qquad \text{(E-8.b)}$$

where

$$S_k = \begin{bmatrix} T_k \\ \beta_{k+1} e_k^T \end{bmatrix}$$

If we define $t_k = \beta_1 e_1 - U_{k+1} S_k y_k$, then $r_k = U_{k+1} t_k$, and $\|r_k\| = \|t_k\|$, assuming U_{k+1} has been computed by exact arithmetic. Hence it is natural to solve the least square problem

$$\min_{y_k} \|\beta_1 e_1 - T_k y_k\| \tag{E-9}$$

which is the basis for USYMLQ. Computationally it is advantageous to solve this problem by the standard LQ factorization (see Paige and Saunders ref. 42).

In this section, we have given a brief discussion of the motivation behind the algorithm USYMLQ. We shall go into greater details concerning the theoretical and computational aspect of the algorithm.

E.2 THE VECTORS OF ALGORITHM 2

In section E.2.1, we shall discuss the properties of the vectors u_k and v_k generated by Algorithm 2; in section E.2.2, we shall discuss the possible choices for the initial vectors u_1 and v_1; in section E.2.3, we shall discuss in detail the LQ factorization. The arguments used for the proofs of the theorems in these sections are based on basic linear algebra tools such as linear independence, minimum polynomial, etc. These sections are included for the sake of completeness.

E.2.1 PROPERTIES OF THE VECTORS U_k AND V_k

This section describes the theory governing the vectors u_k and v_k generated by Algorithm 2.

We define the symbol $\langle x_1, x_2, \ldots, x_n \rangle$ as the vector space spanned by the vectors $x_1, x_2, \ldots, x_n$. The following theorem summarizes the properties of u_k and v_k which are generated by Algorithm 2. It is proved by induction.

Theorem 1. The following four statements are true:

1) $U_k^* U_k = I$
   $V_k^* V_k = I$

2) u_{2k} is a linear combination of vectors in the vector space
   $U_{2k} = \langle b, AA^*b, \ldots, (AA^*)^{k-1}b, Ac, AA^*c, \ldots, (AA^*)^{k-1}Ac \rangle$
   u_{2k-1} is a linear combination of vectors in the vector space
   $U_{2k+1} = \langle b, AA^*b, \ldots, (AA^*)^{k}b, Ac, AA^*c, \ldots, (AA^*)^{k-1}Ac \rangle$
   v_{2k} is a linear combination of vectors in the vector space
   V_{2k}
   $= \langle c, A^*Ac, \ldots, (A^*A)^{k-1}c, A^*b, A^*Ab, \ldots, (A^*A)^{k-1}A^*b \rangle$
   v_{2k+1} is a linear combination of vectors in the vector space
   V_{2k+1}
   $= \langle c, A^*Ac, \ldots, (A^*A)^{k}c, A^*b, A^*Ab, \ldots, (A^*A)^{k-1}A^*b \rangle$

3) Let $A = \hat{U}D\hat{V}^*$ be the singular value decomposition of A; that is, $\hat{U}$ and $\hat{V}$ are orthonormal matrices, and D is a positive and real diagonal matrix: $D = \mathrm{diag}(d_1, \ldots, d_n)$ and let

$$b = \sum_{i=1}^{n} h_i \hat{u}_i,$$

$$\sum_{i=1}^{n} k_i \hat{v}_i,$$

where the h_i and the k_i are scalars, $\hat{u}_i$ is the i-th column of the matrix $\hat{U}$ and $\hat{v}_i$ is the i-th column of the matrix $\hat{V}$. If $b = c$, then the following is true:

$$U_{2k} = \langle \sum_{i=1}^{n} h_i\hat{u}_i,\ \sum_{i=1}^{n} d_i^2 h_i\hat{u}_i, \ldots,\ \sum_{i=1}^{n} d_i^{2(k-1)} h_i\hat{u}_i,\ \sum_{i=1}^{n} d_i k_i\hat{u}_i, \ldots,\ \sum_{i=1}^{n} d_i^{2(k-1)+1} k_i\hat{u}_i \rangle$$

$$U_{2k+1} = \langle \sum_{i=1}^{n} h_i\hat{u}_i,\ \sum_{i=1}^{n} d_i^2 h_i\hat{u}_i, \ldots,\ \sum_{i=1}^{n} d_i^{2k} h_i\hat{u}_i,\ \sum_{i=1}^{n} d_i k_i\hat{u}_i, \ldots,\ \sum_{i=1}^{n} d_i^{2(k-1)+1} k_i\hat{u}_i \rangle$$

$$V_{2k} = \langle \sum_{i=1}^{n} k_i\hat{v}_i,\ \sum_{i=1}^{n} d_i^2 k_i\hat{v}_i, \ldots,\ \sum_{i=1}^{n} d_i^{2(k-1)} k_i\hat{v}_i,\ \sum_{i=1}^{n} d_i h_i\hat{v}_i, \ldots,\ \sum_{i=1}^{n} d_i^{2(k-1)+1} h_i v_i \rangle$$

$$V_{2k+1} = \langle \sum_{i=1}^{n} k_i\hat{v}_i,\ \sum_{i=1}^{n} d_i^2 k_i\hat{v}_i, \ldots,\ \sum_{i=1}^{n} d_i^{2k} k_i\hat{v}_i,\ \sum_{i=1}^{n} d_i h_i\hat{v}_i, \ldots,\ \sum_{i=1}^{n} d_i^{2(k-1)+1} h_i\hat{v}_i \rangle$$

Also if x_k is generated by (E-7.a) and (E-7.b), then the k-th residual $r_k = b - Ax_k$ is in U_{k+1}. We use the symbols u_k and v_k (without the hats) to denote the orthonormal vectors generated by Algorithm 2, and u_k and v_k (with the hats) to denote the singular vectors of the matrix A.

4) If x_k is computed by equation (E-7), and the residual vector r_k is as defined by equation (E-8), then $r_k \neq 0$ implies $u_{k+1} \neq 0$ and $v_{k+1} \neq 0$.

Note that parts 1) and 2) of the Theorem 1 described the "Lanczos-like" characteristics of Algorithm 2. Part 3) of Theorem 1 implies that the asymptotic convergence rate of the conjugate-gradient-type method based on Algorithm 2 is dependent on the square root of the condition number of A rather than the condition number itself, as in the case of solving the normal equation $A^*Ax = A^*b$ by the classical conjugate gradient method. Part 4) takes care of the case when Algorithm 2 terminates, that is, when u_k or v_k are zero vectors; this part of the theorem asserts that in the case when the algorithm terminates, we have arrived at the solution.

We called the procedure of computing the u_k's and the v_k's by Algorithm 2 and the x_k's by equations (E-7.a) and (E-7.b) USYMLQ.

Proof of Theorem 1

1) We prove the following statements by induction:

$$(u_i,u_j) = 0 \quad \text{for } i<j \qquad \text{(E-10.a)}$$
$$(v_i,v_j) = 0 \quad \text{for } i<j \qquad \text{(E-10.b)}$$
$$(Av_i,u_j) = 0 \quad \text{for } i<j+1 \qquad \text{(E-10.c)}$$
$$(A^*u_i,v_j) = 0 \quad \text{for } i<j-1 \qquad \text{(E-10.d)}$$

When $i = 1$, the statement is true by construction. Suppose the statement is true for $i = k$; we show that it is true for $i = k+1$. By Algorithm 2,

$$\beta_{k+1}u_{k+1} = Av_k - \gamma_k u_{k-1} - \alpha_k u_k$$

Forming inner products with u_j on both side of this equation, we obtain

$$\beta_{k+1}(u_{k+1}, u_j) = (Av_k, u_j) - \gamma_k(u_{k-1},u_j) - \alpha_k(u_k,u_j)$$

When $j < k-1$, by the induction hypothesis regarding (E-10.a) and (E-10.c), we see that $(u_{k+1},u_j) = 0$. When $j = k$, the definition of α_k and the induction hypothesis regarding (E-10.a), $(u_{k+1},u_j) = 0$. As for $j = k-1$, we note that the last line of part (b) of Algorithm 2 yields:

$$A^*u_{k-1} = \gamma_k v_k + \beta_k v_{k-2} + \alpha_k{}^* v_{k-1}$$

which again implies:

$$(v_k,A^*u_{k-1}) = \gamma_k(v_k,v_k) + \beta_k(v_k,v_{k-2}) + \alpha_k{}^*(v_k,v_{k-1}).$$

By the induction hypothesis regarding (E-10.b), the equality we just derived yields:

$$(v_k,A^*u_{k-1}) = \gamma_k(v_k,v_k) = \gamma_k$$

The properties of the inner product imply $(v_k,A^*u_{k-1}) = (Av_k,u_{k-1})$. Thus $(u_{k+1},u_j) = 0$ when $j = k-1$. Therefore we have proved (E-10.a). The proof of (E-10.b) is similar and therefore will not be repeated here. By construction $(u_k,u_k) = 1$ and $(v_k,v_k) = 1$ for all k. Therefore the proof of part 1) of Theorem 1 will be completed if we prove (E-10.c) and (E-10.d) above.

Note that Algorithm 2 implies

$$Av_{k+1} = \beta_{k+2}u_{k+2} + \gamma_{k+1}u_k + \alpha_{k+1}u_{k+1},$$

so

$$(Av_{k+1},u_j) = \beta_{k+2}(u_{k+2},u_j) + \gamma_{k+1}(u_k,u_j) + \alpha_{k+1}(u_{k+1},u_j)$$

which equals zero when $j < k$. The proof of (E-10.d) is similar and therefore is omitted here. Thus we have completed the proof of part 1) of Theorem 1.

2) The proof of part 2) of Theorem 1 is also by induction. When $k = 0$, the statement in 2) is true by construction. Now we assume the same statement is true for k and prove that it is true for $k + 1$. Note that for any non-negative integer j,

$$\begin{aligned} U_j &\subseteq U_{j+1} \\ V_j &\subseteq V_{j+1} \\ A^*V_j &\subseteq V_{j+1} \\ AV_j &\subseteq U_{j+1} \end{aligned}$$

Thus $\beta_{k+1}u_{k+1} = Av_k - \alpha_k u_k - \gamma_k u_{k-1} \in U_{k+1}$. Similarly, we can establish the proof regarding v_{k+1}.

3) The singular decomposition $A = \hat{U}D\hat{V}^*$ implies that

$A\hat{V} = D\hat{U}$ and $A^*\hat{U} = D\hat{V}$

which is equivalent to

$A\hat{v}_j = d_j\hat{u}_j$ and $A^*\hat{u}_j = d_j\hat{v}_j$ for $j = 1, 2, .$

Since

$$b = \sum_{j=1}^{n} h_j \hat{u}_j = \sum_{j=1}^{n} k_j \hat{v}_j,$$

$$Ab = \sum_{j=1}^{n} k_j A\hat{v}_j = \sum_{j=1}^{n} k_j d_j \hat{u}_j,$$

and

$$A^*b = \sum_{j=1}^{n} h_j A^*\hat{u}_j = \sum_{j=1}^{n} h_j d_j \hat{v}_j$$

Then

$$A^*Ab = \sum_{j=1}^{n} k_j d_j A^*\hat{u}_j = \sum_{j=1}^{n} k_j d_j^2 \hat{v}_j$$

and

$$AA^*b = \sum_{j=1}^{n} h_j d_j A d_j \hat{v}_j = \sum_{j=1}^{n} h_j d_j^2 \hat{u}_j$$

Also

$$(A^*A)^k b = \sum_{j=1}^{n} k_j d_j^{2k} \hat{v}_j$$

and

$$(AA^*)^k b = \sum_{j=1}^{n} h_j d_j^{2k} \hat{u}j$$

Also note that

$$(A^*A)^k A^*b = A^*(AA^*)^k b = \sum_{j=1}^{n} h_j d_j^{2k} A^*\hat{u}_j = \sum_{j=1}^{n} h_j d_j^{2k+1} \hat{v}_j$$

and

$$(AA^*)^k Ab = A(A^*A)^k b = \sum_{j=1}^{n} h_j d_j^{2k} A\hat{v}_j = \sum_{j=1}^{n} h_j d_j^{2k+1} \hat{u}_j$$

Substituting all these equalities into part 2) of Theorem 1 will yield part 3).

4) To prove part 4), we are going to show that if either u_{k+1} or v_{k+1}, the (k + 1)-th orthonormal vector generated by Algorithm 2 is zero; then $r_k = 0$. Let U_k and V_k be nxk matrices whose columns are respectively the u_i's and the v_i's in the first k steps of Algorithm 2, and T_k is the tridiagonal matrix which was defined above equation (E-6.a) and (E-6.b). If we multiply both sides of equation (E-6.a) on the right by the matrix U_k, then

$$U_k^* A V_k = T_k$$

Let r_k be defined in equation (E-8); then

$$U_k^* r_k = U_k^*(b - Ax_k).$$

Since x_k is as defined by equation (E-7) and $b = \beta_1 U_k e_1$ by Algorithm 2, the above equality can be rewritten as

$$\begin{aligned} U_k{}^* r_k &= U_k{}^*(U_k \beta_1 e_1 - AV_k y_k) \\ &= \beta_1 e_1 - U_k{}^* AV_k y_k \\ &= \beta_1 e_1 - T_k y_k \end{aligned}$$

which is equal to 0 by construction. Thus $(u_j, r_k) = 0$ for $j < k$. However, equation (E-8) implies that

$$U_{k+1}{}^* r_k = U_{k+1}{}^*(U_{k+1}{}^*(\beta_1 e_1 - S_k y_k)) = \beta_1 e_1 - S_k y_k = t_k.$$

If $u_{k+1} = 0$, then $t_k = 0$. However as we have pointed out in a remark following equation (E-7), $\|r_k\| = \|t_k\|$. Therefore $t_k = 0$ implies $r_k = 0$.

We shall now consider the case $v_{k+1} = 0$. If we define x_k and y_k by

$$T_k{}^* y_k = \beta_1 e_1 \qquad \text{(E-11.a)}$$
$$x_k = U_k y_k \qquad \text{(E-11.b)}$$

then $\hat{x}_k$ will be an approximation to the solution of $A^* \hat{x} = b$. Using an argument similar to the one used for the case $u_{k+1} = 0$, we can show that if $v_{k+1} = 0$, then $\hat{r}_k = b - A^* \hat{x}_k = 0$. However, we are not interested in $\hat{r}_k$, and we really want to show that $r_k = 0$.

Note that $\hat{r}_k = 0$ means the solution $\hat{x}$ for $A^* \hat{x} = b$ is in U_k, and $r_k = 0$ means that the solution x for $Ax = b$ is in V_k. We shall show that $\hat{x}$ in U_k implies x in V_k. Let us first write these solution vectors in terms of the singular vectors and singular values of A. Recall that

$$b = \sum_{i=1}^{n} h_i \hat{u}_i = \hat{U} h$$

$$\sum_{i=1}^{n} k_i \hat{v}_i = \hat{V} k$$

where h is a vector whose entries are $h_1, h_2, \ldots$, and k is a vector whose entries are $k_1, k_2, \ldots$. Note that

$$x = A^{-1} b = \hat{V} D^{-1} \hat{U}^* b = \hat{V} D^{-1} \hat{U}^* U h = \hat{V} D^{-1} h = \sum_{i=1}^{n} d_i^{-1} h_i \hat{v}_i.$$

Similarly,

$$x = (A^*)^{-1} b = \sum_{i=1}^{n} d_i^{-1} k_i \hat{u}_i$$

For simplicity, let us assume k even and replace k by 2k. $\hat{x} \in U_{2k}$ means

$$\begin{aligned} \hat{x} &= \sum_{i=1}^{n} d_i^{-1} k_i \hat{u}_i \\ &= \sum_{i=1}^{n} \Big(\sum_{j=0}^{k-1} m_j d_i^{2j} h_i + \sum_{j=0}^{k-1} n_j d_i^{2j+1} k_i \Big) \hat{u}_i \end{aligned}$$

Since the $\hat{u}_i$'s are linearly independent, this means that for $i = 1, 2, \ldots,$

$$d_i^{-1} k_i = \Big(\sum_{j=0}^{k-1} m_j d_i^{2j} h_i + \sum_{j=0}^{k-1} n_j d_i^{2j+1} k_i \Big)$$

Multiply both sides of the above equation by d_i, and move the left hand side to the right hand side; we obtain

$$0 = \left(\sum_{j=0}^{k-1} m_j d_i^{2j+1} h_i + \sum_{j=0}^{k-1} \underline{n}_j d_i^{2j} k_i \right)$$

where $\underline{n}_0 = -1$, and $\underline{n}_j = n_{j+1}$ for $j = 1, 2, \ldots$.

Multiply both the above equations on the right by $d_i \hat{u}_i$ and add all the equations together:

$$0 = \sum_{i=1}^{n} \left(\sum_{j=0}^{k-1} m_j d_i^{2j} h_i + \sum_{j=0}^{k-1} \underline{n}_j d_i^{2j+1} k_i \right) \hat{u}_i$$

The vector on the right hand side of the above equation is in U_{2k+1}. This equality implies that the basis vectors of U_{2k+1} as defined in part 3) of Theorem 1 are linear dependent. Therefore, we can express the first basic vector $\sum_{i=1}^{n} h_i \hat{u}_i$ as a linear combination of the other basis vectors:

$$\sum_{i=1}^{n} h_i \hat{u}_i = \sum_{i=1}^{n} \left(\sum_{j=1}^{k-1} m_j d_i^{2j} h_i + \sum_{j=0}^{k-1} n_j d_i^{2j+1} k_i \right) \hat{u}_i$$

The linear independence of the vectors u_i implies that for each $i = 1, 2, \ldots,$

$$h_i = \left(\sum_{j=1}^{k-1} m_j d_i^{2j} h_i + \sum_{j=0}^{k-1} n_j d_i^{2j+1} k_i \right).$$

Multiply both sides of the equation by $d_i^{-1} v_i$ and sum each side to obtain

$$\sum_{i=1}^{n} d_i^{-1} \hat{v}_i h_i = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} m_j d_i^{2j+1} h_i + \sum_{j=1}^{k-1} n_j d_i^{2j} k_i \right) \hat{v}_i$$

Note that the left hand side is the solution vector for $Ax = b$, and the right hand side is a vector in V_{2k}. The conjugate gradient algorithm can be considered as choosing x_{2k} in V_{2k} so that the residual norm is the least. Therefore, we conclude that $r_{2k} = 0$ when $\hat{v}_{2k+1} = 0$.

The proof for k odd is similar and therefore is omitted here.

The proof of part 4) of Theorem 1 yields the following corollary which is useful in defining a stopping criteria for our algorithm:

Corollary: $\|r_k\| = \|S_k y_k - \beta 1 e\|$, where r_k and S_k are as defined in equation (E-8).

E.2.2 CHOICE OF THE INITIAL VECTORS U_1 AND V_1

Choosing u_1 such that $\beta_1 u_1 = b$ is reasonable. Such a choice is made in the standard CG-type algorithms like SYMMLQ and LSQR. However, we know of no works in the literature to provide us with guidelines or motivation regarding the choice of v_1. The proof of part 4) of Theorem 1 shows that a solution procedure for the equation $A^*x = c$ is "implicitly" embedded in Algorithm 2, combined with equations (E-7.a) and (E-7.b), and this implicit procedure occurs simultaneously with our explicit procedure for $Ax = b$. The proof of part 4) of the theorem also implies that we want to choose c such that we do not solve $A^*x = b$ before we solve $Ax = b$. This is why we cannot choose $c \alpha A^*b$, because the solution vector, which happens, for $A^*x = c$, to be b, lies in U_1, $A^*x = c$ is solved (implicitly) in 1 step, Algorithm 2 terminates and we are not even close to the solution for $Ax = b$. In the proof of part 4) of Theorem 1, we also showed that if we choose $c = b$, then $A^*x = c$ cannot be solved (implicitly) before $Ax = b$ is solved. In other words the choice of $c = b$ guarantees that our algorithm will arrive at the solution. However, this is not the optimal choice. The reason for this is given in the following theorem.

Theorem 2. If the matrix A has m distinct singular values, then the number of iterations USYMLQ takes to converge is bounded by min(2*m,n), while the number of iterations required by the classical CG applied to the normal equation is bounded by min(m,n).

Proof of Theorem 2

From part 3) of Theorem 1, we see that

$$V2m = \langle c, A^*Ac, \ldots, (A^*A)^{k-1}c, A^*b, A^*Ab, \ldots, (A^*A)^{m-1}A^*b \rangle$$

$$= \langle \sum_{i=1}^{n} k_i\hat{v}_i, d_i^2 k_i\hat{v}_i, \ldots, \sum_{i=1}^{n} d_i^{2(m-1)} k_i\hat{v}_i,$$

$$\sum_{i=1}^{n} d_i h_i\hat{v}_i, \ldots, \sum_{i=1}^{n} d_i^{2(m-1)+1} h_i v k_i \rangle$$

which is obtained after 2m steps, contains the solution vector. As for applying the classical CG to the normal equations, we see that the solution vector is in

$$\tilde{V}_m = \langle \sum_{i=1}^{n} k_i\hat{v}_i, \sum_{i=1}^{n} d_i^2 k_i\hat{v}_i, \ldots, \sum_{i=1}^{n} d_i^{2(m-1)} k_i\hat{v}_i \rangle$$

Basis vectors for $\tilde{V}_m$ are generated within k steps of the classical CG applied to the normal equation. Thus the theorem is proved.

We note that the solution vector is contained in the vector space:

$$W_k = \langle \sum_{i=1}^{n} k_i\hat{v}_i, \sum_{i=1}^{n} d_i k_i\hat{v}_i, \sum_{i=1}^{n} d_i^2 k_i\hat{v}_i, \ldots, \sum_{i=1}^{n} d_i^{(m-2)} k_i\hat{v}_i, \sum_{i=1}^{n} d_i^{(m-1)} k_i\hat{v}_i \rangle$$

Note $b = \sum_{i=1}^{n} h_i\hat{u}_i$, for some h_i's. If we could choose c such that $c = \sum_{i=1}^{n} h_i\hat{v}_i$, then $\tilde{V}_k = W_k$, and the number of iterations required by USYMLQ will be bounded by min(m,n). With an argument similar to the proof used for part 4) of theorem 4, we can show that we will not solve $A^*x = c$ before we solve $Ax = b$. However, in practice, we do not want to compute the singular values and singular vectors of the matrix A. Although we can express c in terms of A, A*, and b, namely, $c = A^{-1}(AA^*)^{1/2}b$, we know of no computationally efficient way of computing this ideal c.

The above discussion regarding this ideal c proves the following theorem:

Theorem 3

If $c = A^{-1}(AA^*)^{1/2}b$, and if u_k and v_k are as defined in Algorithm 2, and x_k is as defined in equations (E-7.a) and (E-7.b), then $r_k \neq 0$ implies $u_k \neq 0$, and $v_k \neq 0$; also the number of iterations for this algorithm to converge is bounded by min(m, n) where m is the number of distinct singular values of A, and n is the order of the matrix A.

E.2.3 THE LQ FACTORIZATION

In this section, we describe the solution of equations (E-7.a) and (E-7.b) in detail. As we remarked before, it is computationally advantageous to solve this system by the standard LQ factorization on T_k. The LQ factorization on T_k takes the form:

$$S_k Q_k = \begin{bmatrix} T_k \\ \\ \beta_k \rho^*_k \end{bmatrix} \cdot Q_k = L_k = \begin{bmatrix} \rho_1 & & & \\ \delta_1\rho_2 & & & \\ \xi_1\delta_2\rho_3 & & & \\ \xi_2\delta_3 & \ddots & & \\ \xi_3 & \ddots & & \\ & & \rho_{k-1} & \\ & & \delta_{k-1}\bar{\rho}_k & \\ & & \xi_{k-1}\bar{\delta}_k & \end{bmatrix} \tag{E-12}$$

where $Q_k^* = Q_{2,1}.Q_{3,2} \ldots Q_{k,k-1}$ is a product of plane rotation designed to eliminate the superdiagonal elements $\gamma_1, \gamma_2, \ldots$. Therefore, the left hand side of equation (E-7.a) can be written as

$$T_k y_k = L_k Q_k = \beta_1 e_1$$

We note that y_k has no elements in common with y_{k-1}. We are actually not interested in y_k, but rather it is x_k as defined in equation (E-7.b) that we are after. We want to organize the computation of x_k so that only the most recent iterates need to be saved. The scheme we discuss below avoids the explicit computation of y_k. We let $z_k = Q_k y_k$, $W_k = V_k Q_k$. Thus $x_k = W_k z_k$. We will show that z_k and w_k can be computed by simple recursions.

If we let $\underline{z}_k = Q_k, y_k$, then the first (k-2) entries of $\underline{z}_k$ are the same as those of $\underline{z}_{k-1}$. That is, if we write $\underline{z}_k = (z_1, z_2, \ldots, z_{k-1}, z_k)^T$, then

$$z_{k-1} = (-\xi_{k-3}\, z_{k-3} - \delta_{k-2}\, z_{k-2})/\, \rho_{k-1} \tag{E-13.a}$$

$$\bar{z}_k = (-\xi_{k-2}\, z_{k-2} - \delta_{k-1}\, z_{k-1})/\, \rho_k \tag{E-13.b}$$

Note $\bar{z}_k$ will be replaced by z_k in the next iteration.

Now consider the plane rotation $Q_{k,k-1}$. $Q_{k,k-1}$ operates on the (k-1)-th and k-th row of T_k to eliminate γ_k. This gives the following simple recursion:

$$\begin{bmatrix} \bar{\rho}_{k-1} & \gamma_k \\ \delta_{k-1} & \alpha_k \\ 0 & \beta_k \end{bmatrix} \begin{bmatrix} c_k & s_k \\ s_k & -c_k \end{bmatrix} = \begin{bmatrix} \rho_{k-1} & 0 \\ \delta_{k-1} & \bar{\rho}_k \\ \xi_{k-1} & \delta_k \end{bmatrix} \tag{E-14}$$

where $\rho_1 = \alpha_1$ and $\delta_1 = \beta_1$, and the scalar c_k and s_k are the nontrivial elements of $Q_{k,k-1}$. The quantity $\bar{\rho}_k$ and $\bar{\delta}_k$ are to be replaced by ρ_k and δ_k in the next iteration. Consider the nxk matrix

$$W_k = V_k Q_k^*$$

Note that the first (k-2) columns of W_k are the same as those of W_{k-1}. That is, if we write

$$W_k = (w_1, w_2, \ldots, w_{k-1}, \bar{w}_k)$$

where the w_i's are the columns of W_k, then

$$w_{k-1} = c_k * \bar{w}_{k-1} + s_k v_k \tag{E-15.a}$$

$$\bar{w}_k = s_k \bar{w}_{k-1} - c_k v_k \tag{E-15.b}$$

Note that $\bar{w}_k$ will be replaced by w_k in the next iteration. Therefore, if we first substitute $z_k = Q_k y_k$, then $W_k = V_k Q_k$ into equation (E-7.b), we get

$$x_k = V_k y_k = V_k Q_k * z_k = W_k z_k$$

If we let $x_0 = 0$,

$$\bar{x}_{k-1} = \bar{x}_{k-2} + w_{k-1} z_{k-1}, \tag{E-16}$$

then

$$x_k = \bar{x}_{k-1} + \bar{w}_k \bar{z}_k \tag{E-17}$$

Note that z_{k-1}, w_{k-1}, and $\bar{x}_{k-1}$ together with $\bar{z}_k$, are computed in the k-th iteration. Also note that z_k is to be replaced in the next iteration. We could have computed x_k according to equation (E-17), but it is not necessary to do so until the last iteration. From the above discussion, we see that only the last two iterates need to be saved.

Now we consider the stopping criteria of our algorithm. Two stopping criteria are implied by part 4) of Theorem 1 as well as Corollary 1:

a) If β_k or γ_k are small enough, r_k will be small enough.
b) However it is possible for r_k to be small when the norms of either u_k or v_k are not small. Corollary 1 permits us to compute the norm of the k-th residual r_k explicitly:

$$\|r_k\| = \|s_k - \beta_1 e_1\| = |s_k z_{k-1} - c_k \bar{z}_k| * \beta_{k+1} \tag{E-18}$$

In summary, at the k-th iterations, the following quantities are computed:

i) The vectors u_k and v_k according to Algorithm 2
ii) c_k and s_k, which are the nontrivial elements of the plane rotation $Q_{k,k-1}$ defined in equations (E-14) and (E-12)
iii) The scalars ρ_{k-1}, δ_{k-1}, $\bar{\xi}_{k-1}$, $\bar{\rho}_k$ and $\bar{\delta}_k$, which are the nonzeroes of the last two columns of L_k defined in equation (E-12)
(iv) z_{k-1} as defined in equation (E-13)
(v) $\bar{w}_{k-1}$ as defined in equation (E-15)
(vi) $\bar{x}_{k-1}$ as defined in equation (E-16)
(vii) $\|r_k\|$ according to equation (E-18).

Note that if the algorithm terminates, we need to compute x_k according to equation (E-17) before returning to the calling program.

Algorithm 3 USYMLQ

1. (Initialize.)
   $\beta_1 u_1 = b, v_1 = u_1, w_1 = v_1, x_0 = 0, \gamma_1 = \beta_1, \rho_1 = \alpha_1, \delta_1 = \beta_1$

2. For i = 1, 2, ..., repeat steps 3 to 7.

3. (Continue the tridiagonalization.)

   $p = Au_i - \gamma_i u_{i-1}$
   $q = A^* u_i - \beta_i^* v_{i-1}$
   $\alpha_i = u^* p$
   $\beta_{i+1} u_{i+1} = p - \alpha_i u_i$
   $\gamma_{i+1} v_{i+1} = q - \alpha_i v_i$

4. (Construct and apply the plane rotation.)
   diagonal element of L_i in row (i-1): $\rho_{i-1} = (\bar{\rho}_{i-1}^2 + \gamma_i^2)/$
   plane rotation: $c_i = \bar{\rho}_{i-1} / \rho_{i-1}$
   $s_i = \gamma_i / \rho_{i-1}$
   subdiagonal element of L_i in row i: $\delta_i = \delta_{i-1} c_i + \gamma_i s_i$
   diagonal element of L_i in row i: $\bar{\delta}_i = \bar{\delta}_{i-1} s_i - \gamma_i c_i$

5. (Update z, x, and w.)

   $z_{i-1} = (-\xi_{i-3} z_{i-3} - \delta_{i-2} z_{i-2}) / \rho_{i-1}$
   $\bar{z}_i = (-\xi_{i-2} z_{i-2} - \delta_{i-1} z_{i-1}) / \bar{\rho}_i$

   $\bar{x}_{i-1} = \bar{x}_{i-2} + z_{i-1}(c_i^* \bar{w}_{i-2} + s_i v_i)$
   $\bar{w}_{i-1} = \bar{w}_{i-1} s_i - v_i c_i$

6. (Test for convergence.)

   If one of the following holds
   (i) β_i is small enough
   (ii) γ_i is small enough

   then $\bar{w}_i = w_{i-1} s_1 - v_i s_i$
   $x_i = \bar{x}_i + z_i \bar{w}_i$
   EXIT

   else go to step 3.

E.3 NUMERICAL EXPERIMENTS

Our new algorithm, USYMLQ (UnSYMmetric LQ factorization) looks very much like Paige and Saunders' LSQR (Least Square QR factorization) (see ref. 40). The theoretical results in sections E.1 and E.2 show that they are indeed different algorithms. In this session, we present numerical experiments which demonstrate that they have different convergence properties.

E.3.1 GENERATION OF PROBLEMS

We generated three test problems to compare the performance of USYMLQ with LSQR. The test problems have coefficient matrices of the form:

$$A = PDQ$$

where P and Q are Householder transformations and D is a complex diagonal matrix. Since P and Q are orthonormal matrices, the condition number of A, say, K, is defined as follows:

$$K = d_{max} / d_{min}$$

where d_{max} and d_{min} are respectively the largest and smallest absolute values of the diagonal entries of D. Moreover, the singular values of the matrix A are just the absolute values of the diagonal entries of D.

We set the i-th entry of the true solution, x, for our test problems to be (n-i) where n is the order of the matrix. The right hand sides, b, are obtained by multiplying the true solution with the coefficient matrix: b = Ax.

E.3.2 CONCLUSION FROM THE NUMERICAL EXPERIMENTS

The numerical results of our experiment are illustrated by the graphs in figures 58, 59, and 60. The x-axis of the figures is the iteration number; the y-axis is the residual norms. The solid lines represent the convergence history of USYMLQ, and the dashed line represents the convergence history of LSQR. N stands for the order of test matrix, K is the condition number, and S is the number of distinct singular values.

Test problem 1 illustrates that USYMLQ can handle an ill-conditioned problem with distinct singular values quite satisfactorily whilst LSQR seems to have trouble. The result of test problem 2 is predicted by Theorem 2; when there are multiple singular values, LSQR converges better. Test problem 3 illustrates the performance of the algorithms for large, well-conditioned matrices. In this case, the convergence rates approach the asymptotic convergence rates.

E.4 THE INCOMPLETE FACTORIZATION

There are many variants of the incomplete factorization of a matrix. The one that we have been using is a product of a unit lower and an upper triangular matrix: M = LU, and M satisfies the following properties:

1. If $a_{i,j} \neq 0$, then $m_{i,j} = a_{i,j}$.

2. L has the same sparsity structure as the lower triangular matrix of A, and U has the same sparsity structure as the upper triangular matrix of A,

where A is the matrix obtained from the finite difference equation of the small disturbance transonic flow potential equation, without the wake terms. In other words, in the two-dimensional case, A is a 5-diagonal matrix. We obtain L and U by starting with the regular Gaussian elimination, and ignoring all the "fill-ins."

As an example in the two-dimensional case, when IMAX = 5 and JMAX = 4, we may write M = LU as in figure 61 with the coefficients defined by Algorithm 4 below:

If we write $A = (a_{i,j})$, then Algorithm 4 describes L and U for the harmonically oscillating flat plate problems:

Algorithm 4 (Incomplete LU factorization)

$d_1 = a_{1,1}$
For i = 1,2,....(IMAX-2)*(JMAX-2)
    $u_i = a_{i,i+1}$
    $q_i = a_{i,i+JMAX-2}$
For i = 2, 3,..., JMAX-2
    $l_1 = a_{i,i-1}/d_{i-1}$
    $d_i = a_{i,i} - l_i * u_{i-1}$
For i = JMAX-1, JMAX,...(IMAX-2)*(JMAX-2)
    $l_i = a_{i,i-1}/d_{i-1}$
    $p_i = a_{i,i-JMAX-2/d_{i-1}}$
    $d_i = a_{i,i} - l_i u_{i-1} - p_i q_{i-JMAX-2}$

Then $R = (r_{i,j}) = A - M$ has one subdiagonal and one superdiagonal of nonzeroes, that is, R is of the form shown in figure 62.

Thus if A is the matrix of the finite difference equation of the small disturbance transonic flow potential equation, then

$$A - M = R + W$$

where W is a matrix with eight nonzero columns contributed by the wake terms.

We apply the conjugate gradient method to an equation of the form:

$$(I + M^{-1}(R + W))x = M^{-1}b$$

REFERENCES

1 Ehlers, F. E.: "A Finite Difference Method for the Solution of the Transonic Flow Around Harmonically Oscillating Wings." NASA CR-2257, January 1974.

2 Weatherill, W. H.; Ehlers, F. E.; and Sebastian, J. D.: "Computation of the Transonic Perturbation Flow Fields Around Two- and Three-Dimensional Oscillating Wings." NASA CR-2599, December 1975.

3 Weatherill, W. H.; Sebastian, J. D.; and Ehlers, F. E.: "The Practical Application of a Finite Difference Method of Analyzing Transonic Flow Over Oscillating Airfoils and Wings." NASA CR-2933, 1978.

4 Ehlers, F. E.; Sebastian, J. D.; and Weatherill, W. H.: "An Investigation of Several Factors Involved in a Finite Difference Procedure for Analyzing the Transonic Flow About Oscillating Airfoils and Wings." NASA CR-159143, 1979.

5 Weatherill, W. H.; and Ehlers, F. E.: "A User's Guide for A344 - A Program Using a Finite Difference Method To Analyze Transonic Flow Over Oscillating Airfoils." NASA CR-159141, 1979.

6 Weatherill, W. H.; Ehlers, F. E.; Yip, E.; and Sebastian, J. D.: "Further Investigation of a Finite Difference Procedure for Analyzing the Transonic Flow About Harmonically Oscillating Airfoils and Wings." NASA CR-3195, May 1980.

7 Ehlers, F. E.; and Weatherill, W. H.: "A Harmonic Analysis Method for Unsteady Transonic Flow and Its Application to the Flutter of Airfoils." NASA CR-3537, May 1982.

8 Rowe, W. S.; Winther, B. A.; and Redman, M. C.: "Prediction of Unsteady Aerodynamic Loadings Caused by Trailing Edge Control Surface Motions in Subsonic Compressible Flow - Analysis and Results." NASA CR-2003, March 1972.

9 Redman, M. C.; Rowe, W. S.; and Winther, B. A.: "Prediction of Unsteady Aerodynamic Loadings Caused by Trailing Edge Control Surface Motions in Subsonic Compressible Flow - Computer Program Description." NASA CR-112015, March 1972.

10 Klunker, E. B.: "Contribution to Methods of Calculating the Flow About Thin Lifting Wings at Transonic Speeds - Analytic Expressions for the Far Field." NASA TN D-6530, 1971.

11 Engquist, B.; and Majda, A.: "Radiation Boundary Conditions for Acoustic and Elastic Wave Calculations." Communications on Pure and Applied Mathematics, Vol. 32, May 1979, pp. 313-357.

12 Murman, E. M.; and Cole, J. D.: "Calculation of Plane Steady Transonic Flow." AIAA Journal, Vol. 9, January 1971, pp. 114-121.

13 Krupp, J. A.; and Murman, E. M.: "Computation of Transonic Flows Past Lifting Airfoils and Slender Bodies." AIAA Journal, Vol. 10, July 1972, pp. 880-887.

14 Stahara, S. S.: "Operational Manual for Two-Dimensional Transonic Code TSFOIL." NASA CR-3064, December 1973.

15 Rizzetta, D. P.; Chin, W. C.: "Effect of Frequency in Unsteady Transonic Flow." AIAA Journal, Vol. 117, July 1979, pp. 779-781.

16 Jou, W-H.; and Murman, E. M.: "A Phenomenological Model for Displacement Thickness Effects of Transonic Shock Wave-Boundary Layer Interactions," AGARD Conference, Colorado Springs, Colo., September 29, 1980.

17 Borland, C.; Rizzetta, D.; Yoshihara, H.: "Numerical Solution of Three-Dimensional Unsteady Transonic Flow Over Swept Wings." AIAA Paper No. 80-1369, July 1980.

18 Yip, E. L.: "FORTRAN Subroutines for Out-of-Core Solution of Large Complex Linear Systems." NASA CR-159142, December 1979.

19 Bland, S. R.: "Development of Low-Frequency Kernel-Function Aerodynamics for Comparison with Time-Dependent Finite-Difference Methods," NASA TM 83283, May 1982.

20 Lessing, H. C.; Troutman, J. L.; and Menees, G. P.: "Experimental Determination of the Pressure Distribution on a Rectangular Wing Oscillating in the First Bending Mode for Mach Numbers From 0.24 to 1.30." NASA TN D-344, December 1960.

21 Rickets, R. H.; Sandford, M. C.; Seidel, D. A.; and Watson, J. J.: "Transonic Pressure Distributions on a Rectangular Wing Oscillating in Pitch." AIAA Paper No. 83-0923. (Presented at the 24th AIAA/ACME/ASCE/AHS Structures, Structural Dynamics and Materials Conference, May 2-4, 1984).

22 Hess, R. W.; Wynne, E. C.; and Cazier, F. W., jr.: "Static and Unsteady Pressure Measurements on a 50 Degree Delta Wing at M = 0.9." AIAA Paper No. 82-0686, May 1982.

23 Guruswamy, G. P. and Goorjian, P. M.: "An Efficient Coordinate Transformation Technique for Unsteady, Transonic Aerodynamic Analysis of Low Aspect-Ratio Wings." AIAA Paper No. 84-0872.

24 Borland, C. J.: "Further Development of XTRAN3S Computer Program." NASA CR 172335, May 1984.

25 Hestenes, M. R.; and Stiefel, E.: "Methods of Conjugate Gradients for Solving Linear Systems." Journal of Research of the National Bureau of Standards, Vol. 49: pp. 409-436, 1952.

26 Reid, J. K.: "On the Method of Conjugate Gradients for the Solution of Large Sparse Sets of Linear Equations." (Ed. J. K. Reid). Academic Press, pp. 231-254, 1971.

27 Hafez, M.; and Wong, Y. S.: "Minimum Residual Method with Newton-like Iterative Procedure for Transonic Flow Calculations." ICASE Report No. 81-30, September 1981.

28 Axelson, O: "Conjugate Gradient Type Methods for Unsymmetric and Inconsistent Systems of Linear Equations." Linear Algebra and its Applications, Vol. 29; pp. 1-16, 1980.

29 Concus, P.; and Golub, G.: "Use of Fast Direct Methods for the Efficient Numerical Solution of Nonseparable Elliptic Equations." SIAM J. Numer. Anal., Vol. 10, pp. 1103-1120, 1973.

30 Elman, H. C.: "Iterative Methods for Large, Sparse Non-symmetric Systems of Linear Equations." Ph.D. Thesis, Yale University, 1982.

31 Rehm, R. G.; and Lewis, J. G.: "The Numerical Solution of a Pressure Equation by Hybrid Conjugate Gradients," Journal of Research, National Bureau of Standards, Vol. 85, No. 5, pp. 367-390.

32 Kershaw, D. S.: "The Incomplete Cholesky Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations." Report UCID-30083, Lawrence Livermore Laboratory, 1976.

33 Chandra, R.: "Conjugate Gradient Methods for Partial Differential Equations." Ph.D. Thesis, Yale University, 1978.

34 Householder, A. S.: "The Theory of Matrices in Numerical Analysis." Blaisdell Publishing Company, 1964.

35 Varga, R. S.: "Matrix Iterative Analysis." Prentice-Hall, 1962.

36 Adam, J; Swarztrauber, P.; and Sweet, R.: "FISHPACK (version 3): A Package of Fortan Subroutines for the Solution of Elliptic Partial Differential Equations." Distributed by the National Center of Atmospheric Research, Boulder, Colo., 1979.

37 Hafez, M. M.; Rizk, M. H.; and Murman, E. M.: "Numerical Solution of the Unsteady Transonic Small-Disturbance Equations." Paper presented at the 44th Meeting of the Structures and Materials Panel of AGARD, Lisbon, Portugal, April 1977.

38 Williams, M. H.: "Linearization of Unsteady Transonic Flows Containing Shocks." AIAA Journal, Vol. 17, p. 394, April 1979.

39 Seebass, A. R.; Yu, N. J.; and Fung, K-Y.: "Unsteady Transonic Flow Computations." Unsteady Aerodynamics, AGARD Conference Proceedings No. 227, September 1977.

40 Paige, C. C.; and Saunders, M. A.: "LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares." ACM Transactions on Mathematical Software, Vol. 8, pp. 43-71, 1982.

41 Bunch, J. R.; and Kaufman, L.: "Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems." Report CU-CS-063-75, Department of Mathematics, University of California at San Diego, 1975.

42 Paige, C. C.; and Saunders, M. A.: "Solution of Sparse Indefinite Systems of Equations and Least Squares Problems." Report CS-399, Department of Computer Science, Stanford University, 1973.

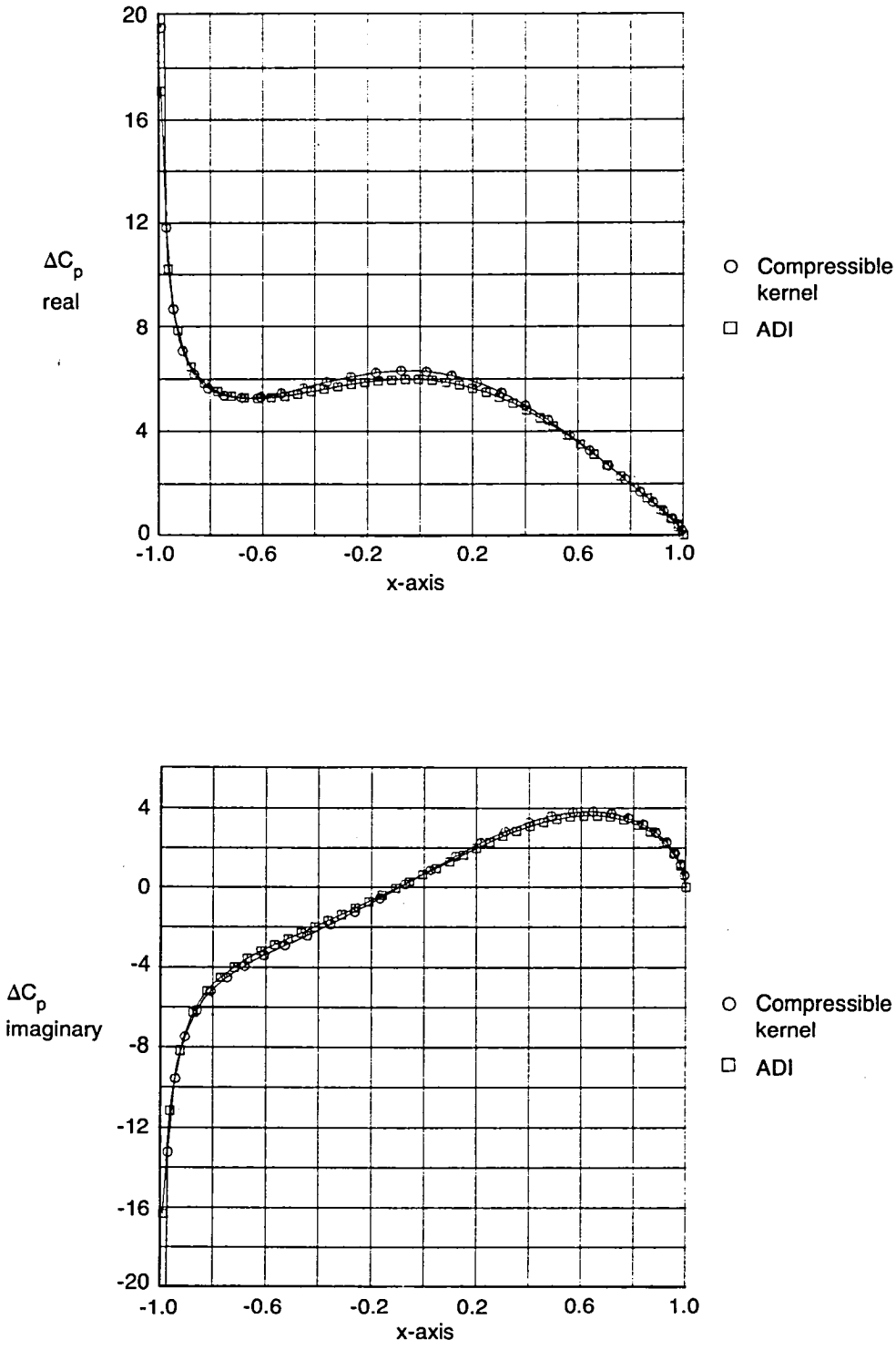


Figure 1.—Comparison of ADI With Kernel Function, Pressure Results, Zero Thickness Airfoil, $M = 0.9$, $k = 0.3$

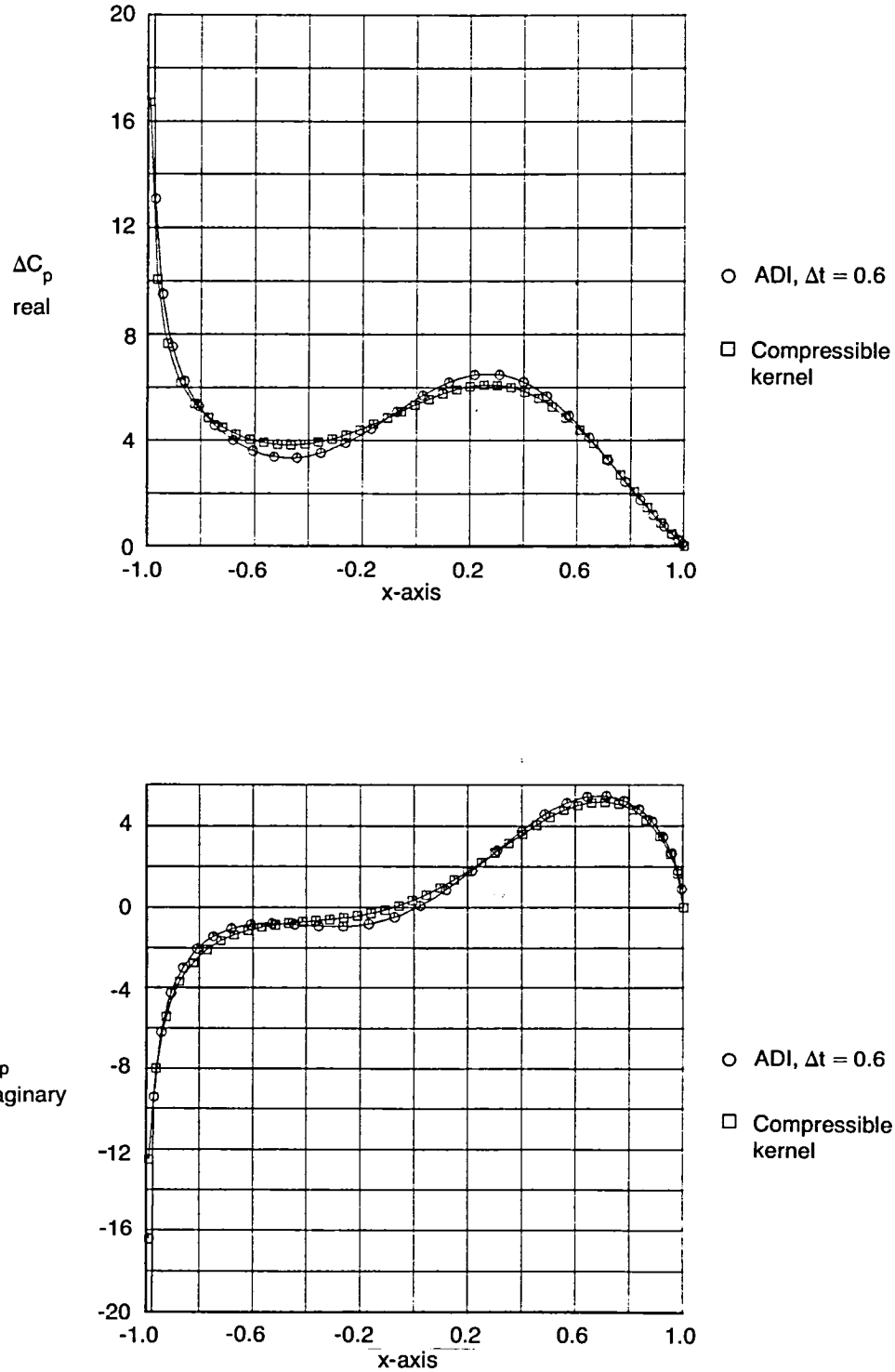


Figure 2.—Comparison of ADI With Kernel Function, Pressure Results, Zero Thickness Airfoil, $M = 0.9$, $k = 0.45$

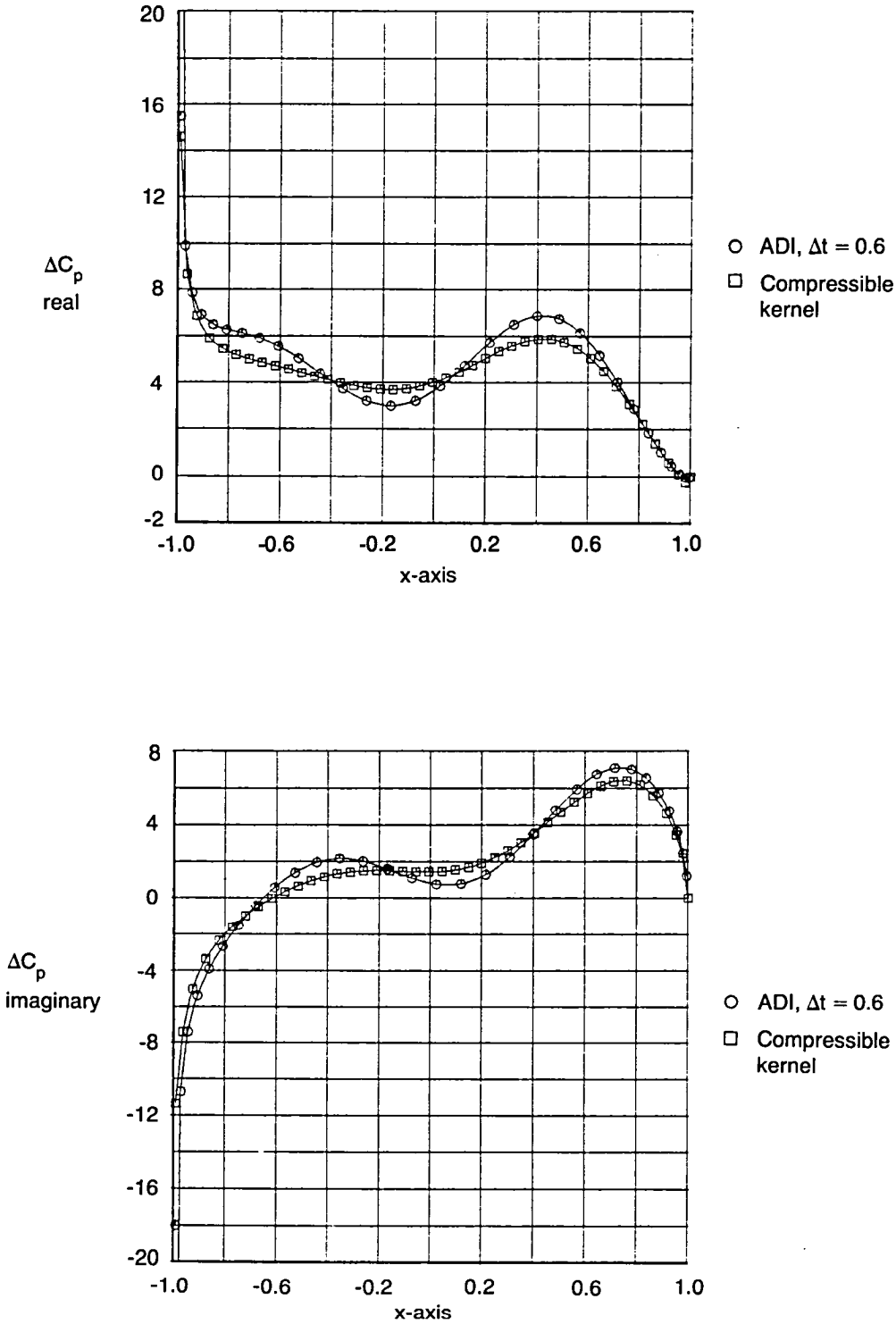


Figure 3.—Comparison of ADI With Kernel Function, Pressure Results, Zero Thickness Airfoil, M = 0.9, k = 0.6

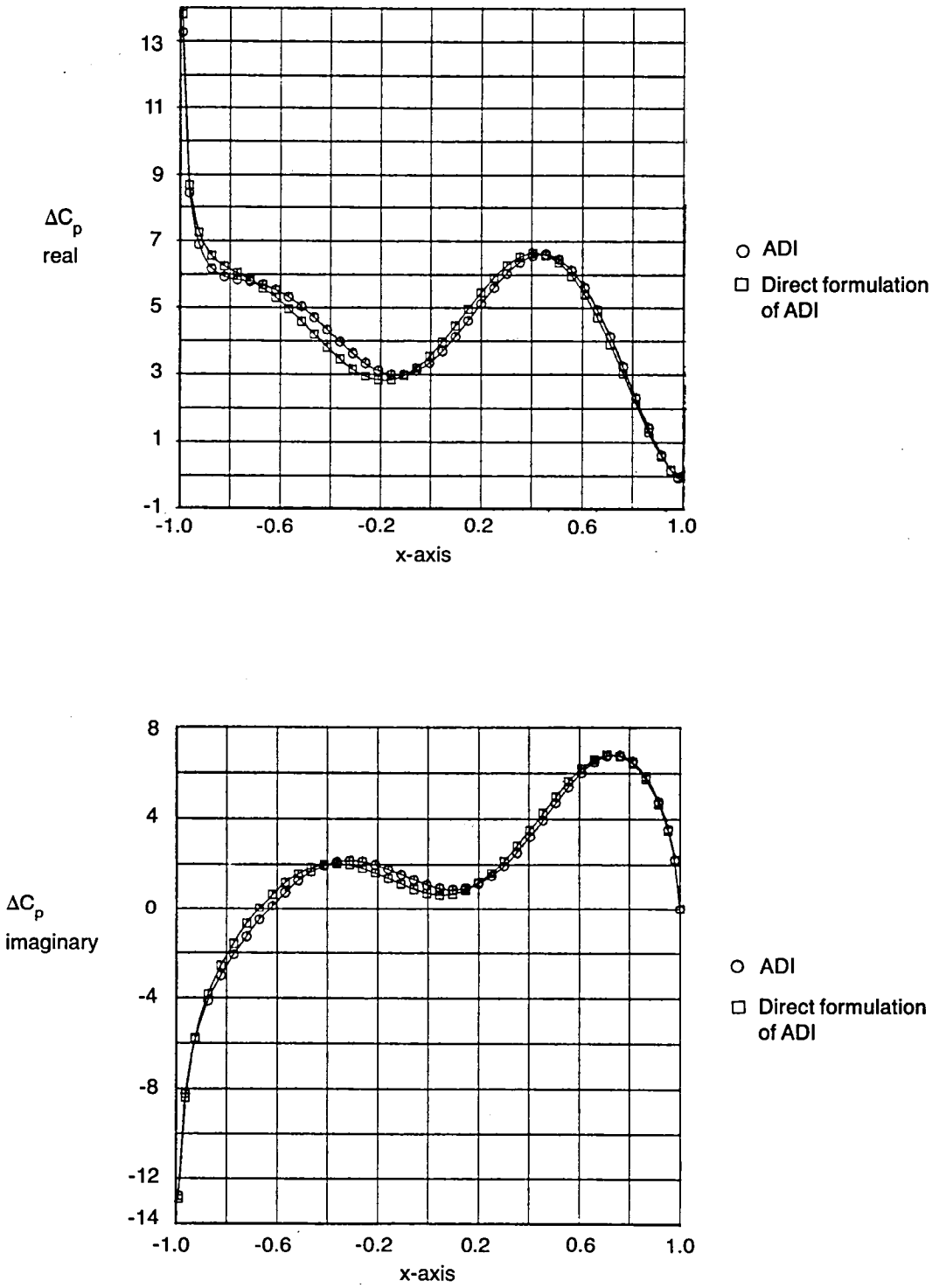


Figure 4.—Comparison of ADI With Direct Solution of Converged ADI Equation, Pressure Results, Zero Thickness Airfoil, M = 0.9, k = 0.6

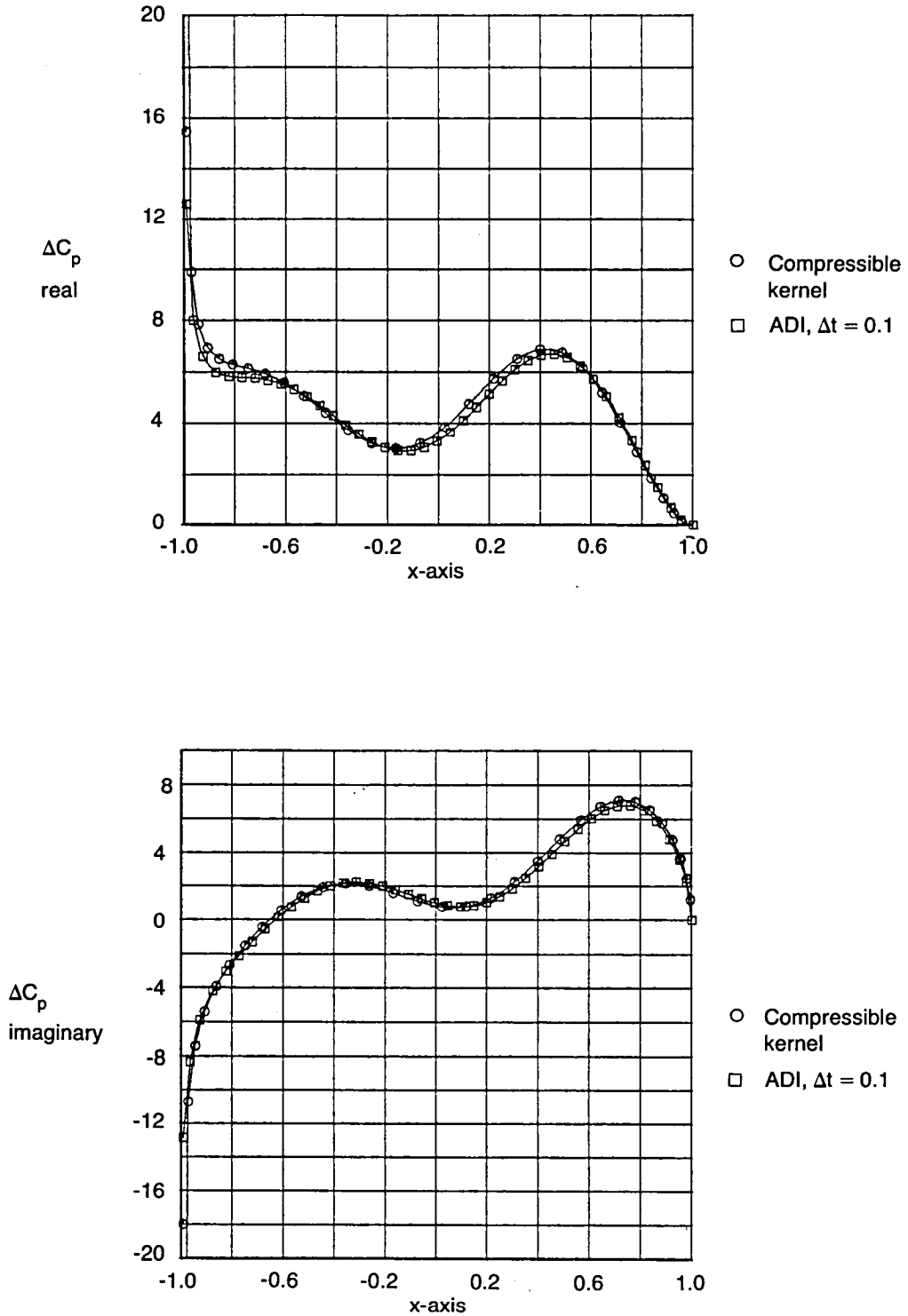


Figure 5.—Comparison of ADI Using Second-Order δx Representation With Kernel Function, Pressure Results, Zero Thickness Airfoil, M = 0.9, k = 0.6

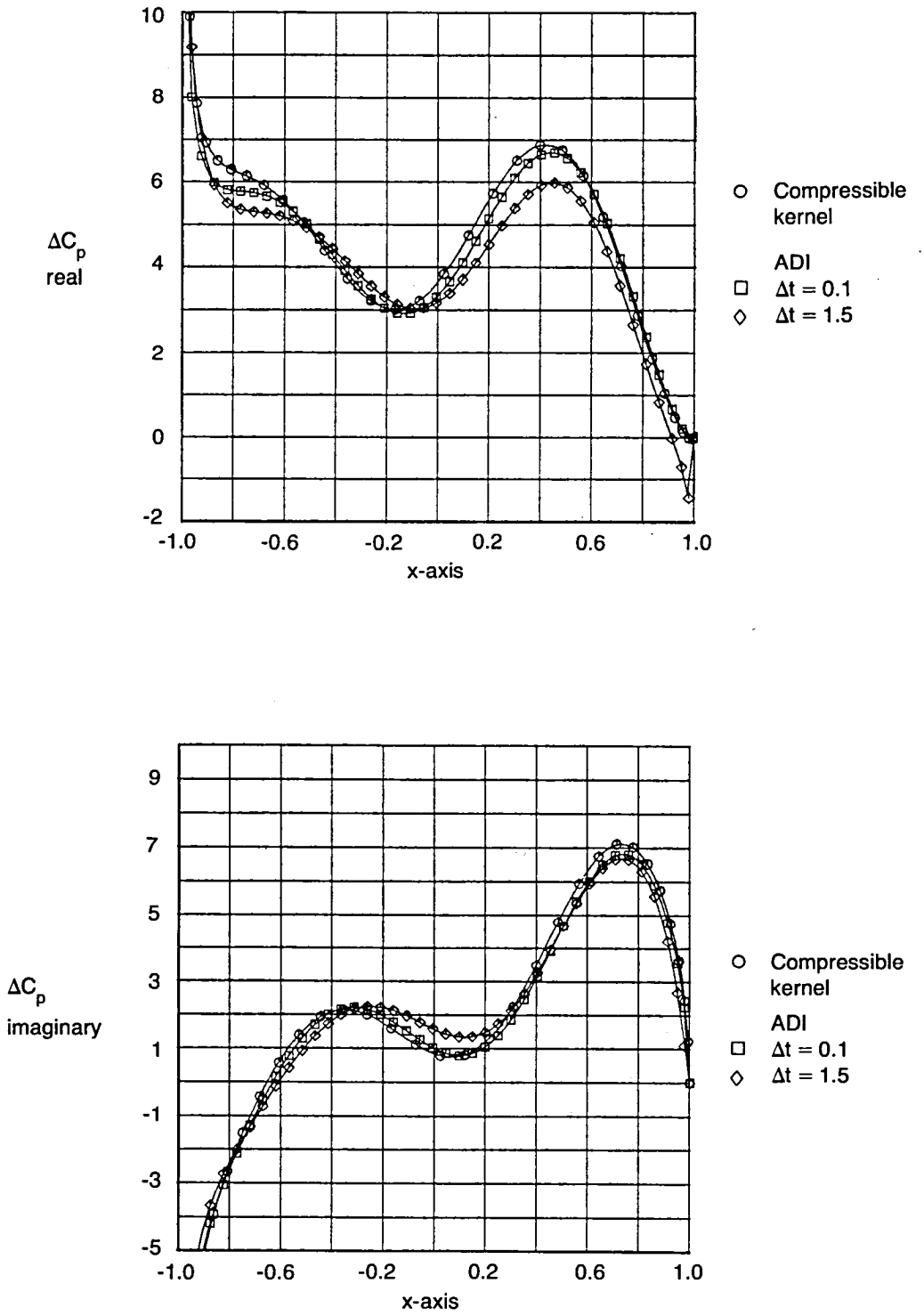


Figure 6.—Comparison of ADI With Kernel Function, Pressure Results for Two Values of Δt, Zero Thickness Airfoil, M = 0.9, k = 0.6

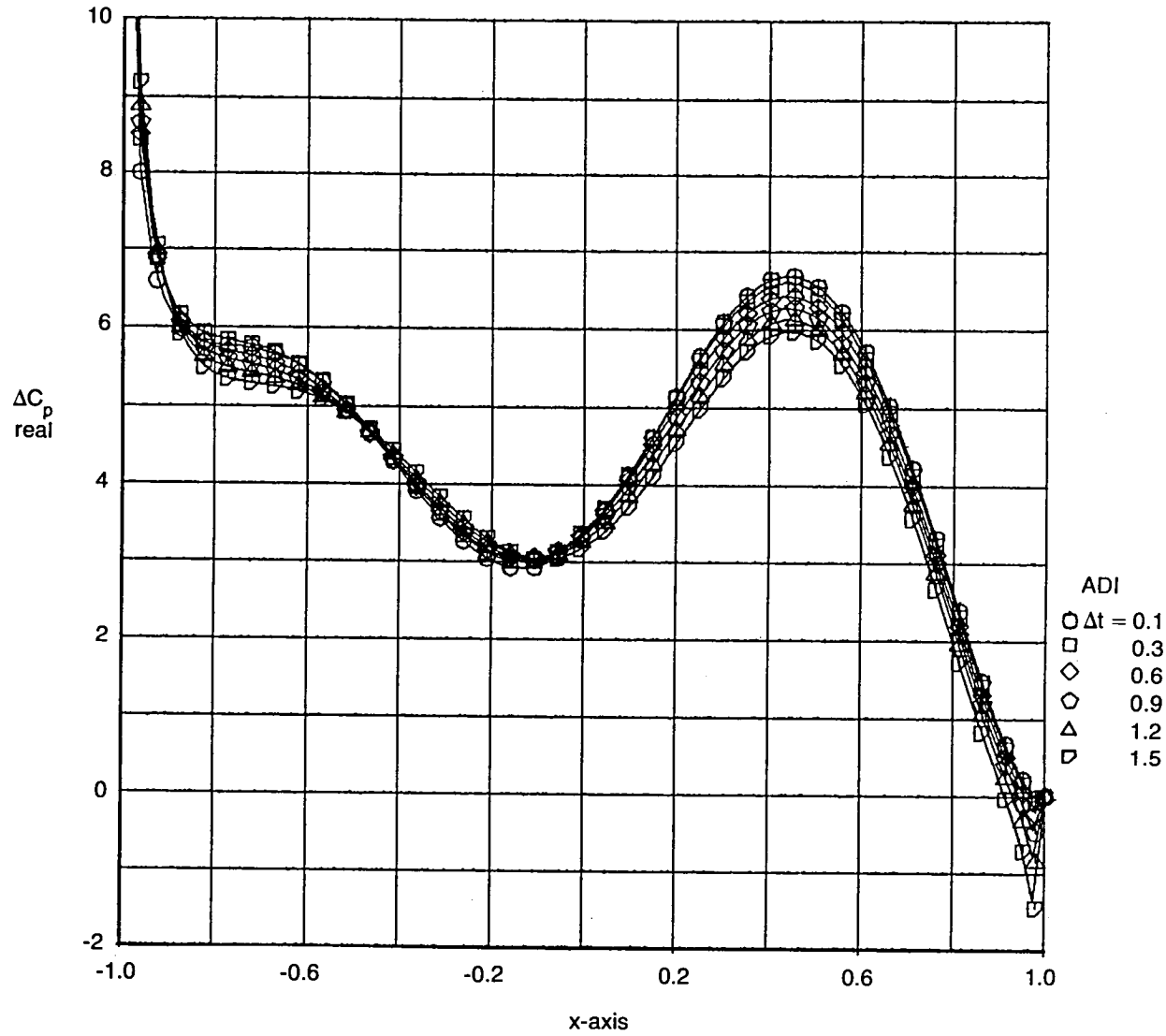


Figure 7.—Comparison at ADI Pressure Results for Several Values of Δt, Zero Thickness Airfoil, M = 0.9, k = 0.6

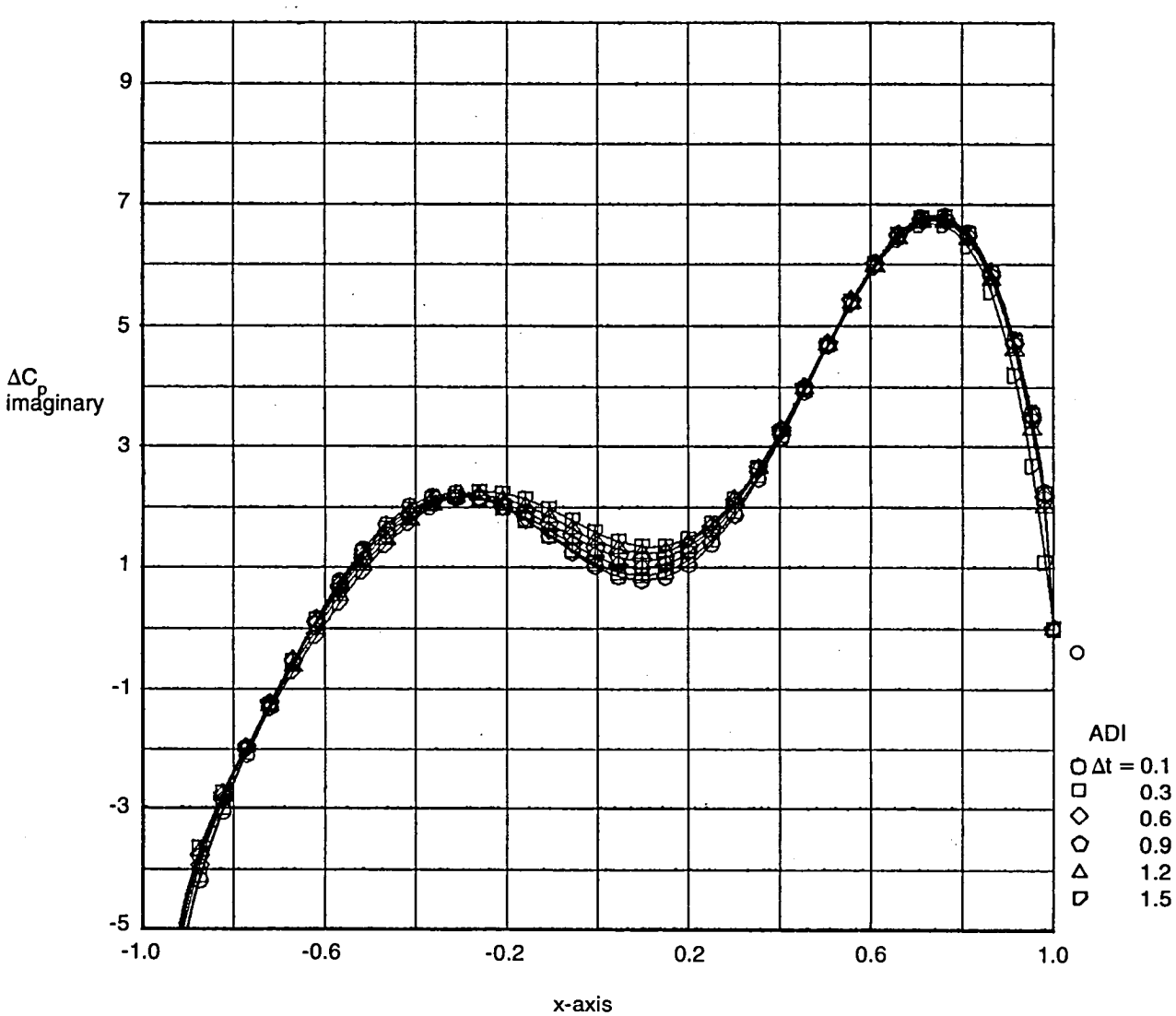


Figure 7.—Comparison at ADI Pressure Results for Several Values of Δt, Zero Thickness Airfoil, M = 0.9, k = 0.6 (Concluded)

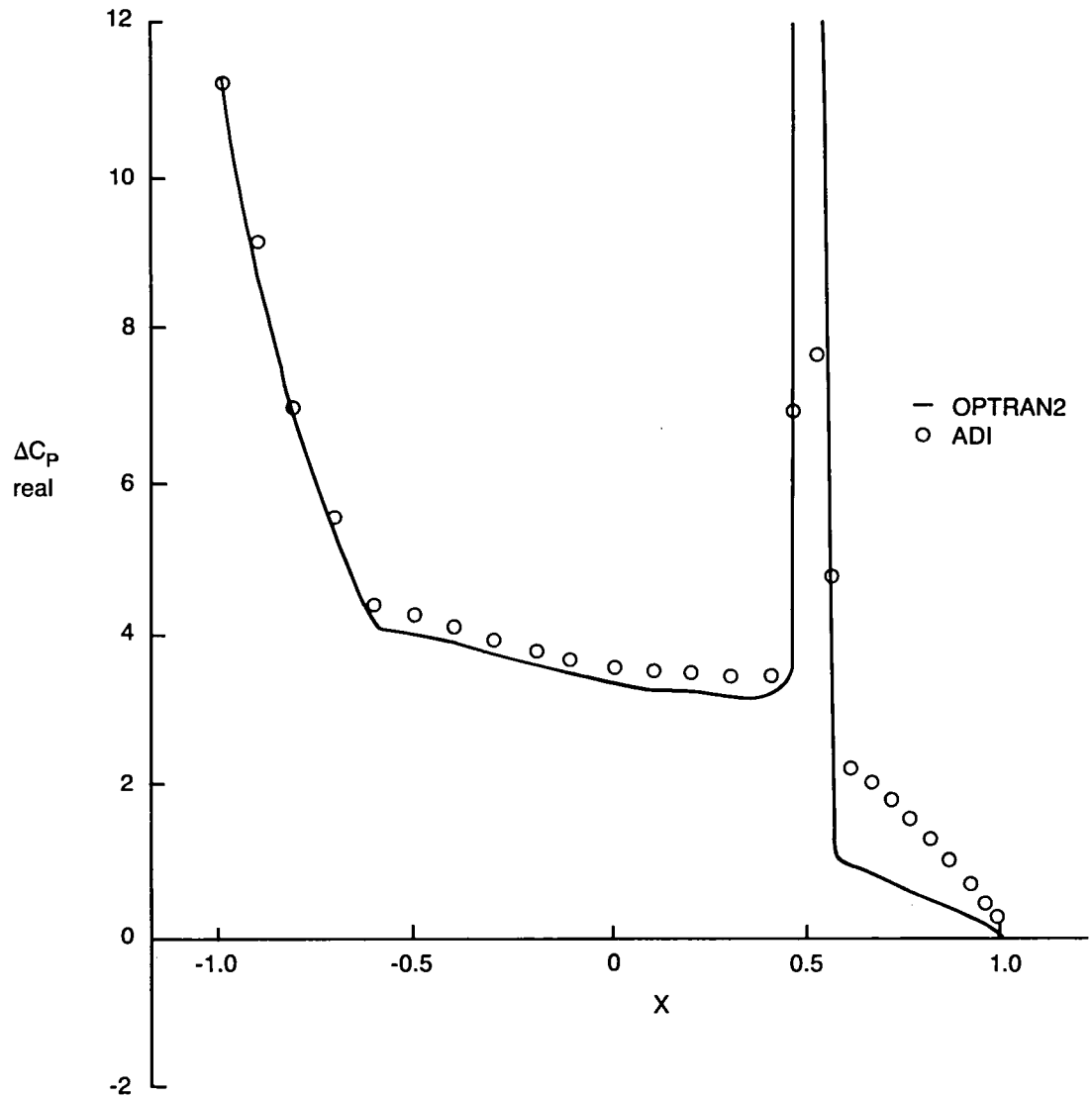


Figure 8.—Comparison of ADI With OPTRAN2, Pressure Results, NACA 64A010 Airfoil, M = 0.85, k = 0.15

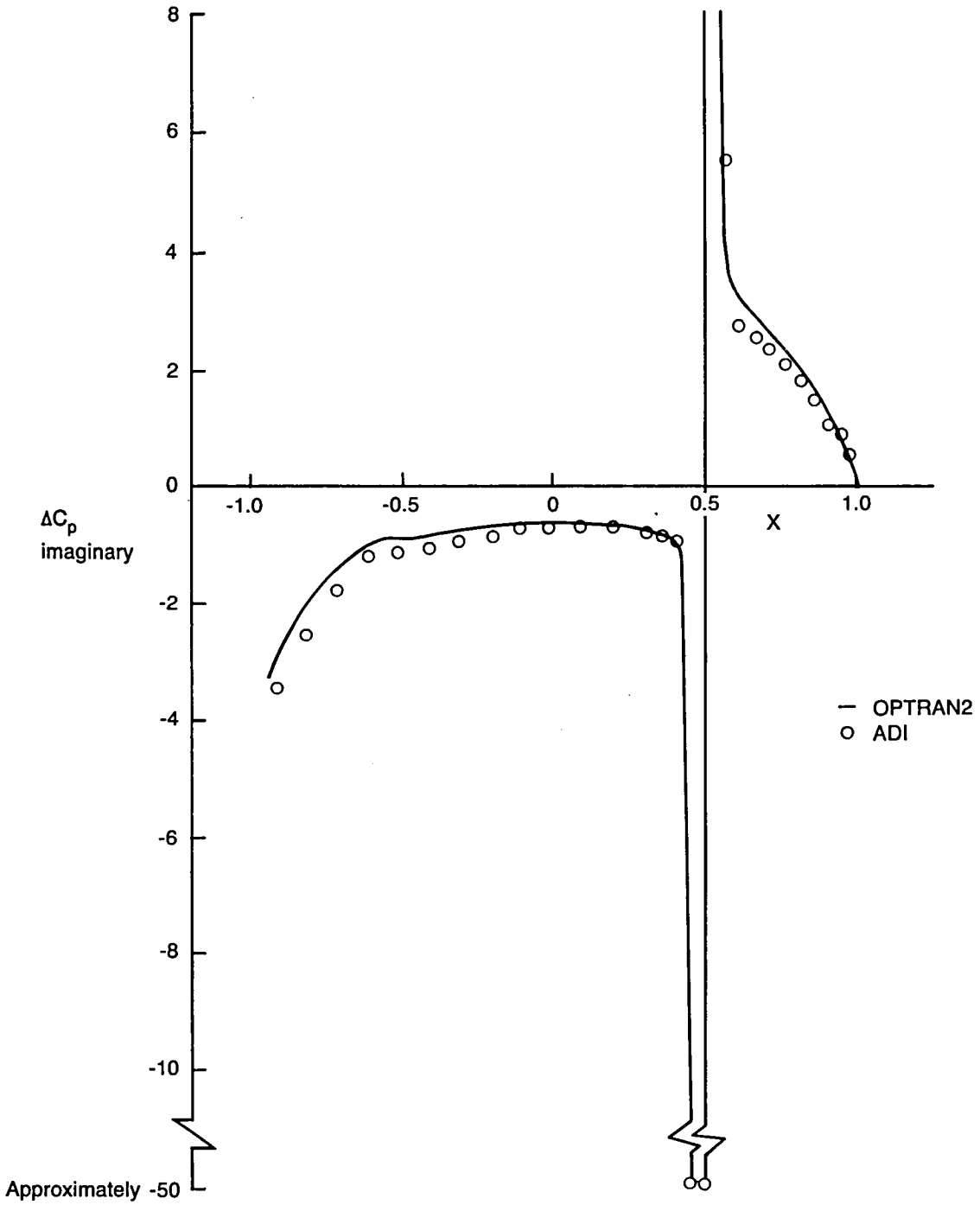


Figure 8.—Comparison of ADI With OPTRAN2, Pressure Results, NACA 64A010 Airfoil, M = 0.85, k = 0.15 (Concluded)

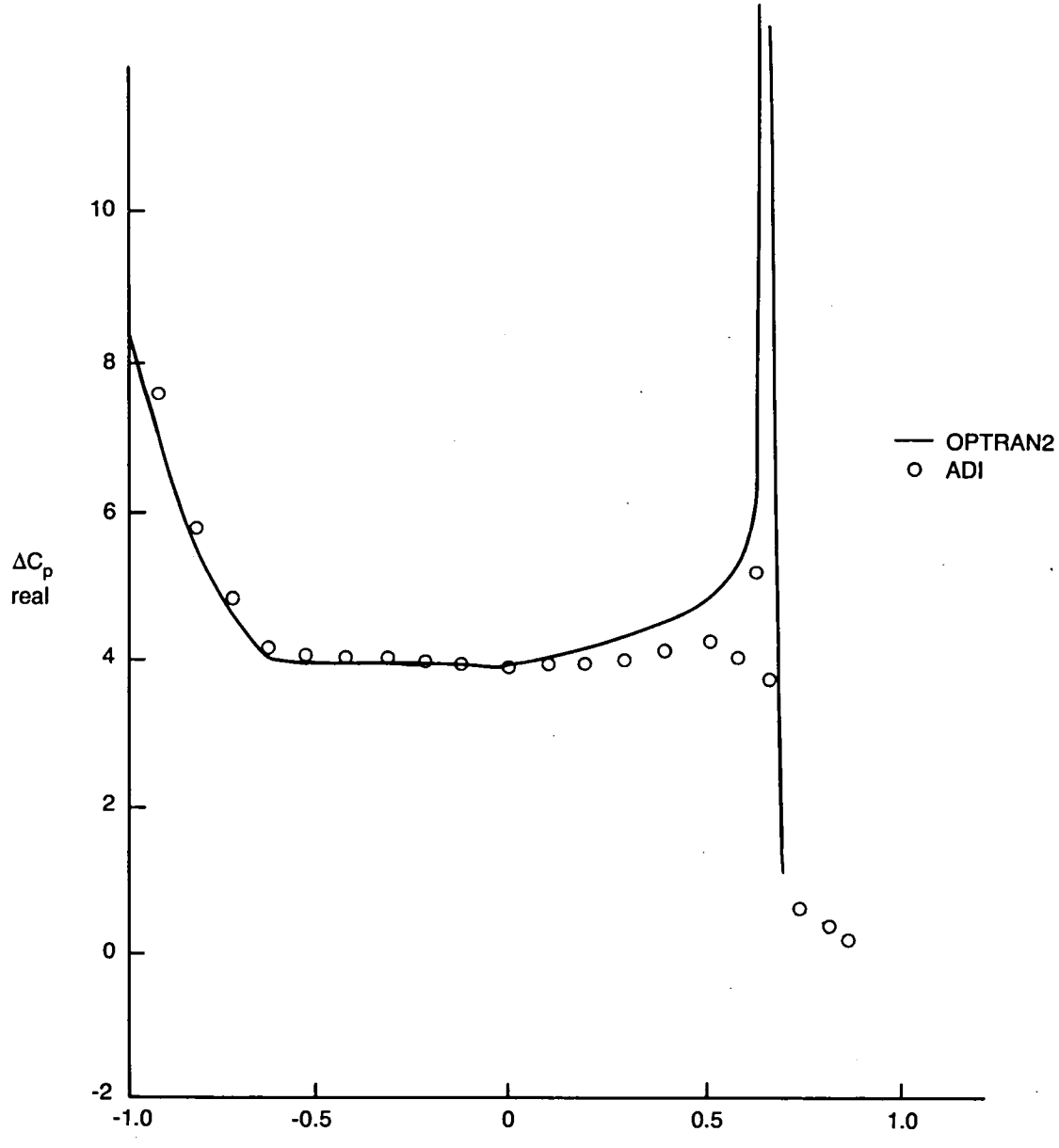


Figure 9.—Comparison of ADI With OPTRAN2, Pressure Results, NACA 64A010 Airfoil, M = 0.86, k = 0.4

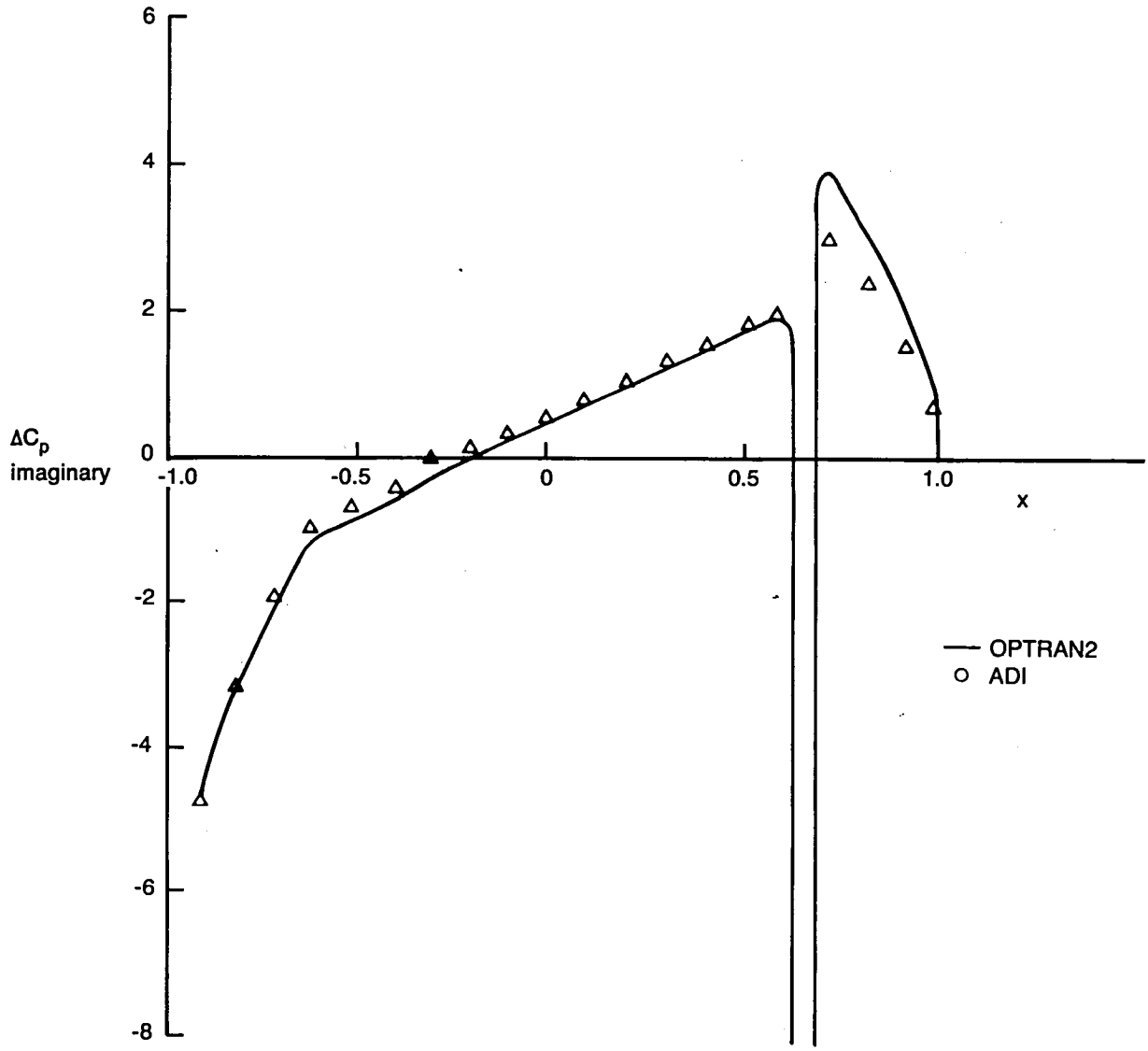


Figure 9.—Comparison of ADI With OPTRAN2, Pressure Results, NACA 64A010 Airfoil, M = 0.86, k = 0.4 (Concluded)

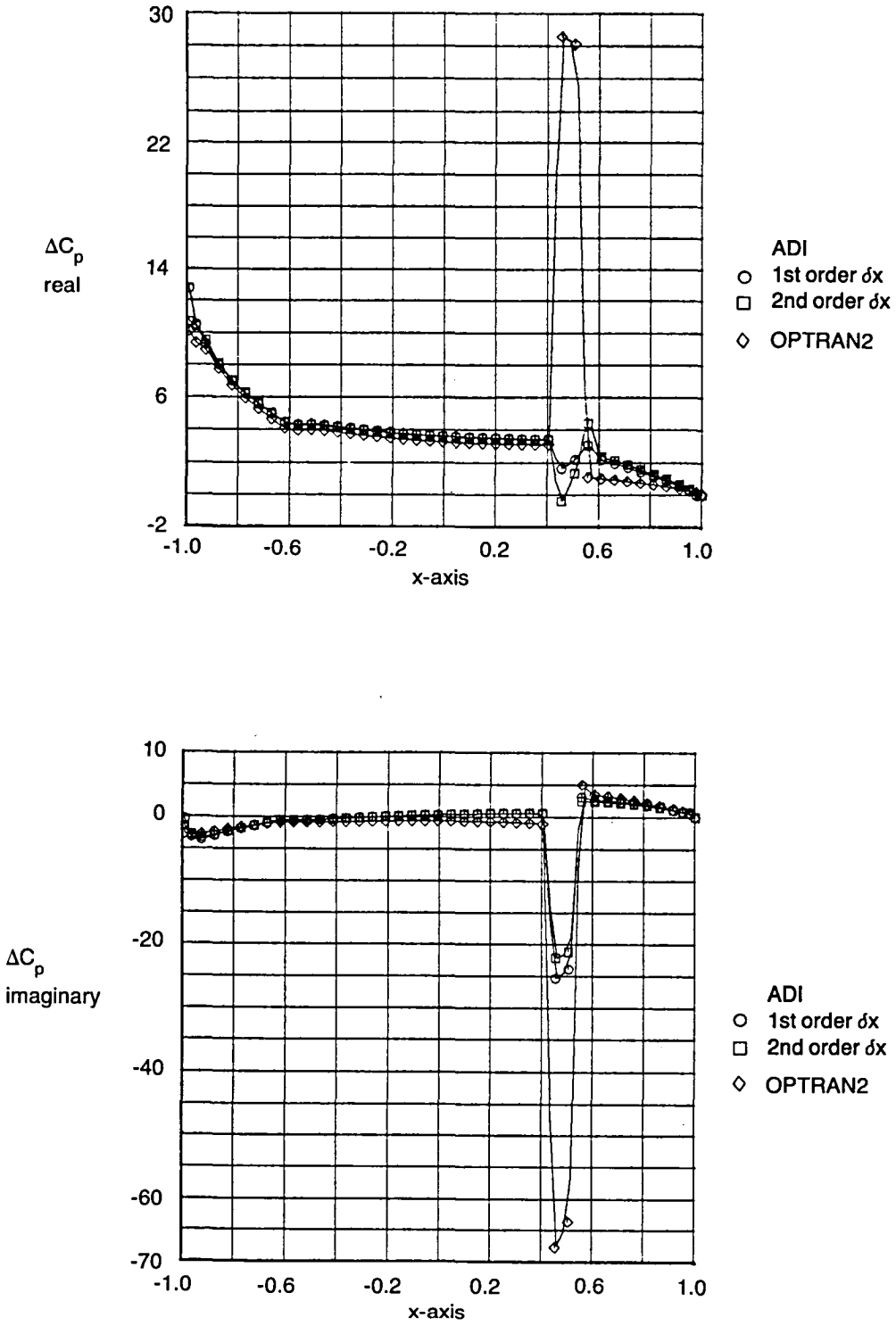


Figure 10.—Comparison of ADI With OPTRAN2, Pressure Results for Two δx Representations, NACA 64A010 Airfoil, M = 0.85, k = 0.25

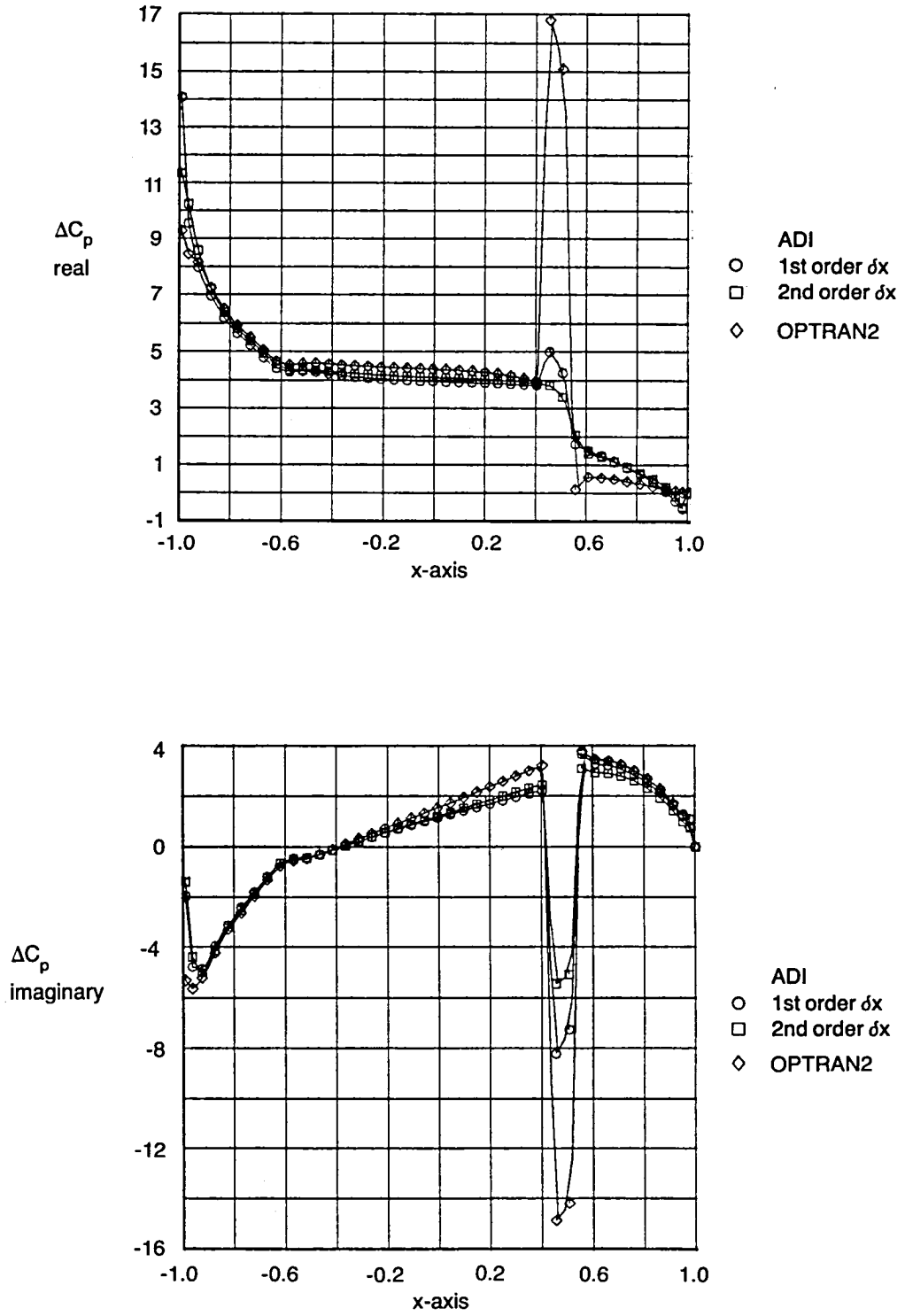


Figure 11.—Comparison of ADI With OPTRAN2, Pressure Results for Two δx Representations, NACA 64A010 Airfoil, M = 0.85, k = 0.4

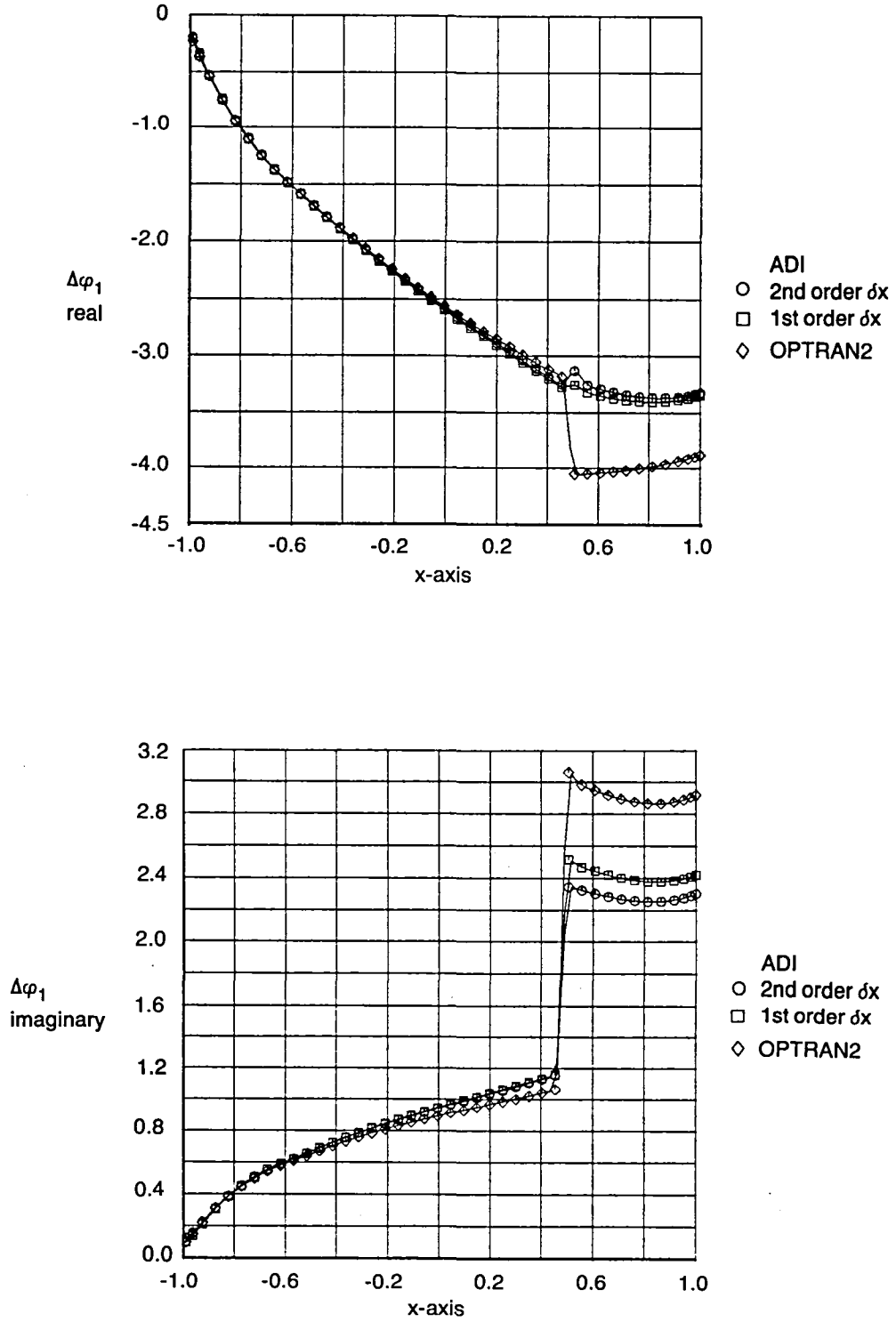


Figure 12.—Comparison of ADI With OPTRAN2, Potential Results for Two δx Representations, NACA 64A010 Airfoil, M = 0.85, k = 0.25

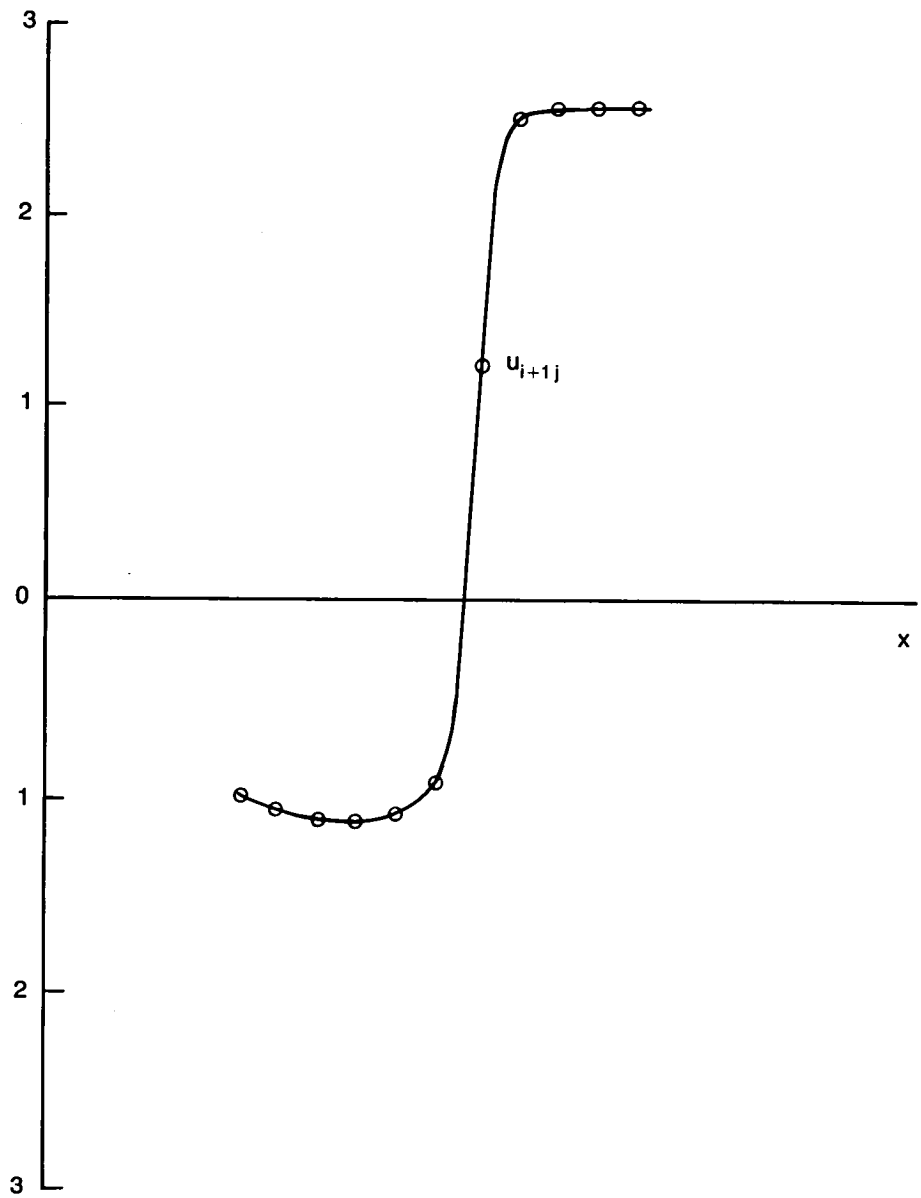


Figure 13.—Example of a Captured Shock, NACA 64A010 Airfoil, M = 0.85

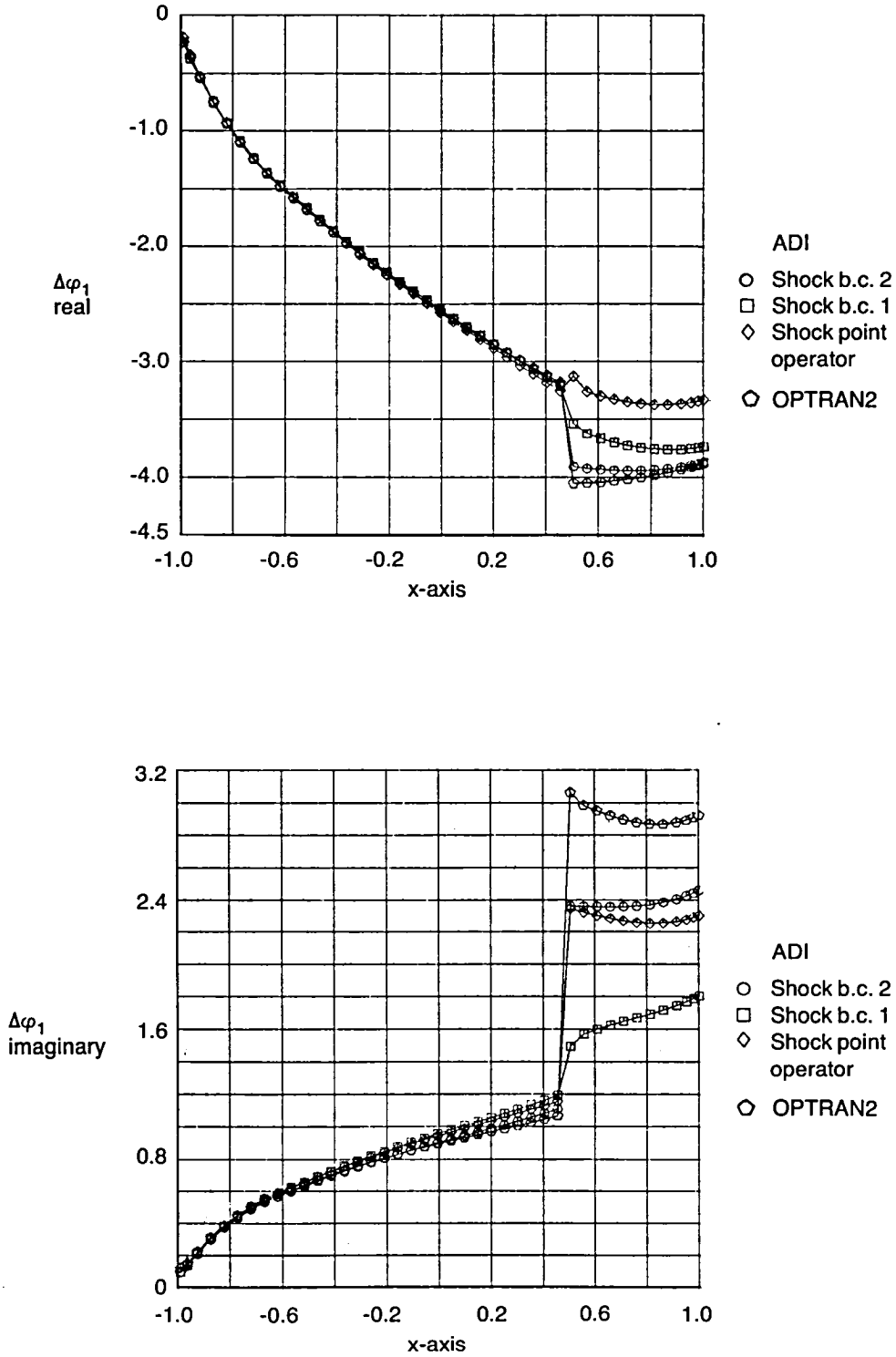


Figure 14.—Comparison of Potential Results From the ADI Using Shock Boundary Conditions With OPTRAN2 Using a Shock Point Operator, NACA 64A010 Airfoil, M = 0.85, k = 0.25

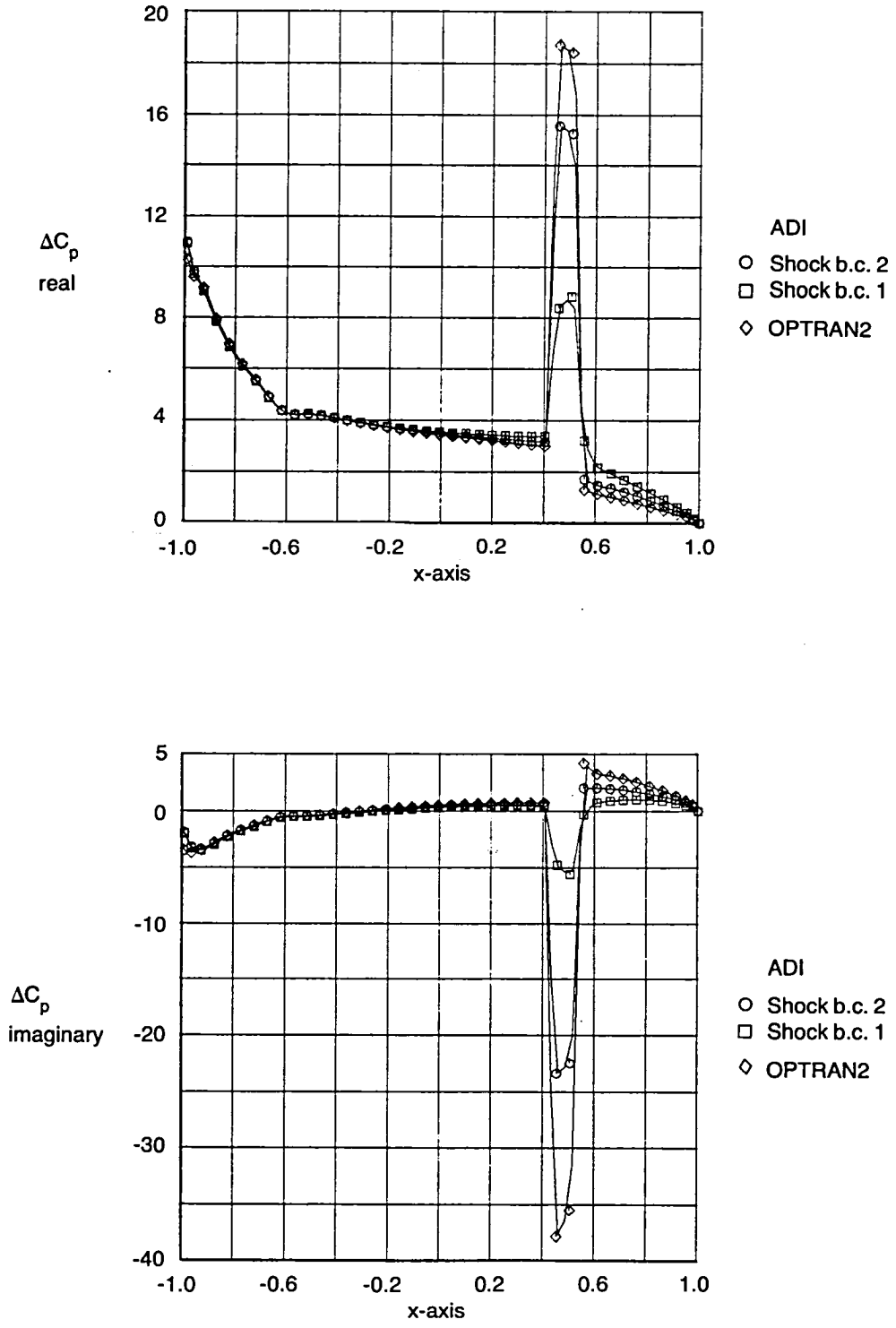


Figure 15.—Comparison of Pressure Results From ADI Using Shock Boundary Conditions With OPTRAN2 Using a Shock Point Operator, NACA 64A010 Airfoil, M = 0.85, k = 0.25

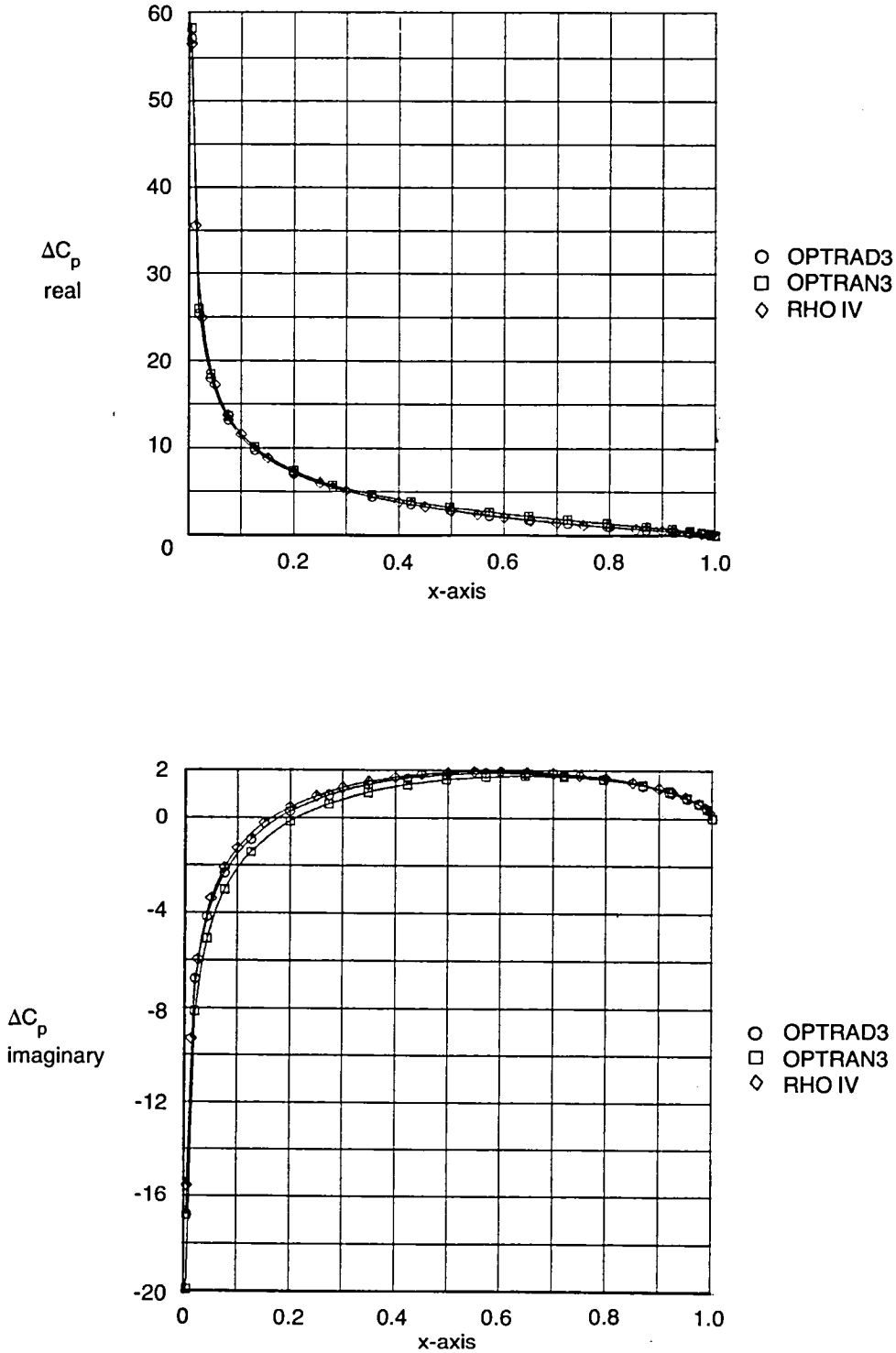


Figure 16.—Comparison of OPTRAD3 and OPTRAN3 With RHOIV, Pressure Distributions, Aspect Ratio 3 Rectangular Wing of Zero Thickness, M = 0.9, k = 0.1, Root Chord

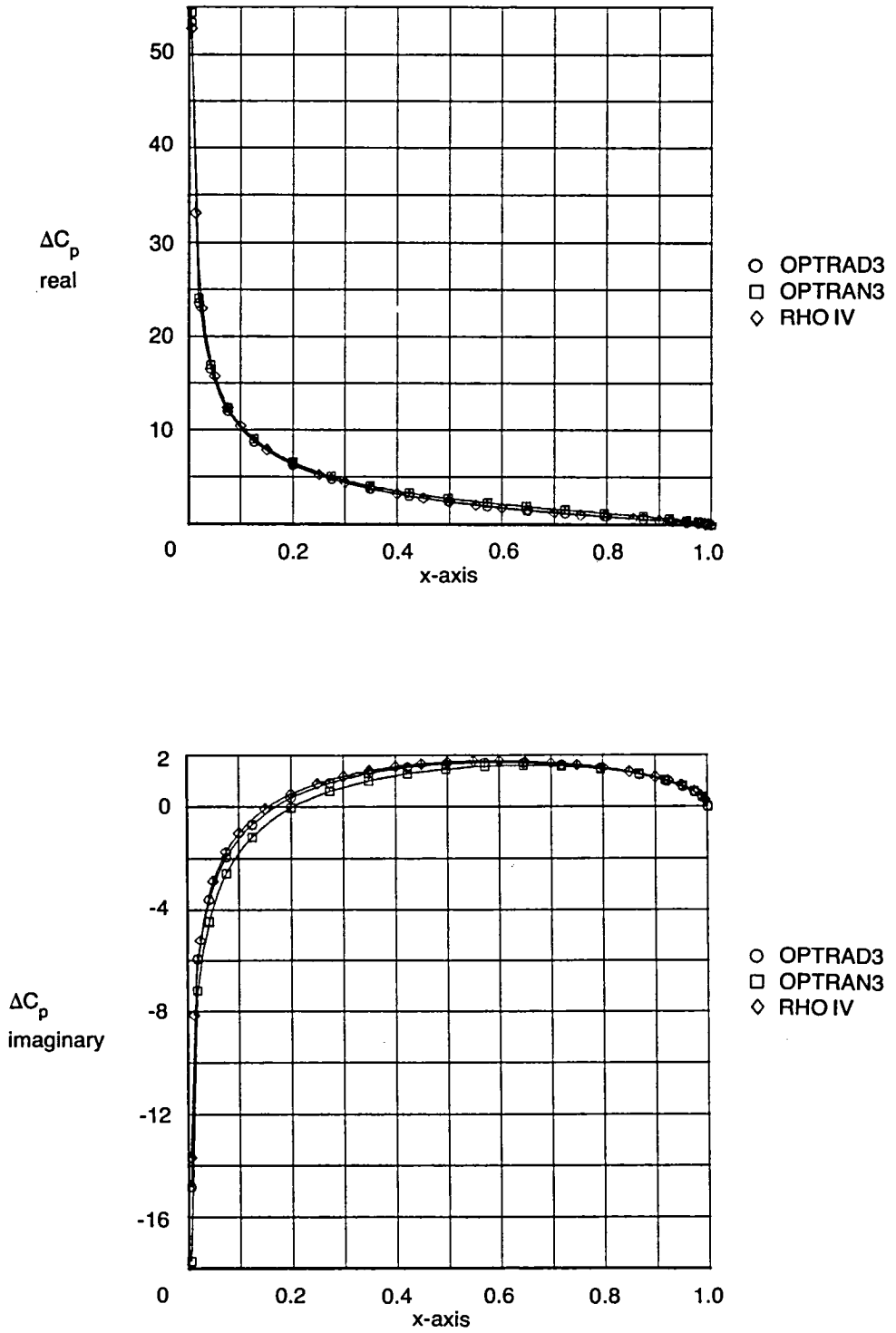


Figure 17.—Comparison of OPTRAD3 and OPTRAN3 With RHOIV, Pressure Distributions, Aspect Ratio 3 Rectangular Wing of Zero Thickness, $M = 0.9$, $k = 0.1$, $\bar{\eta} = 0.46$

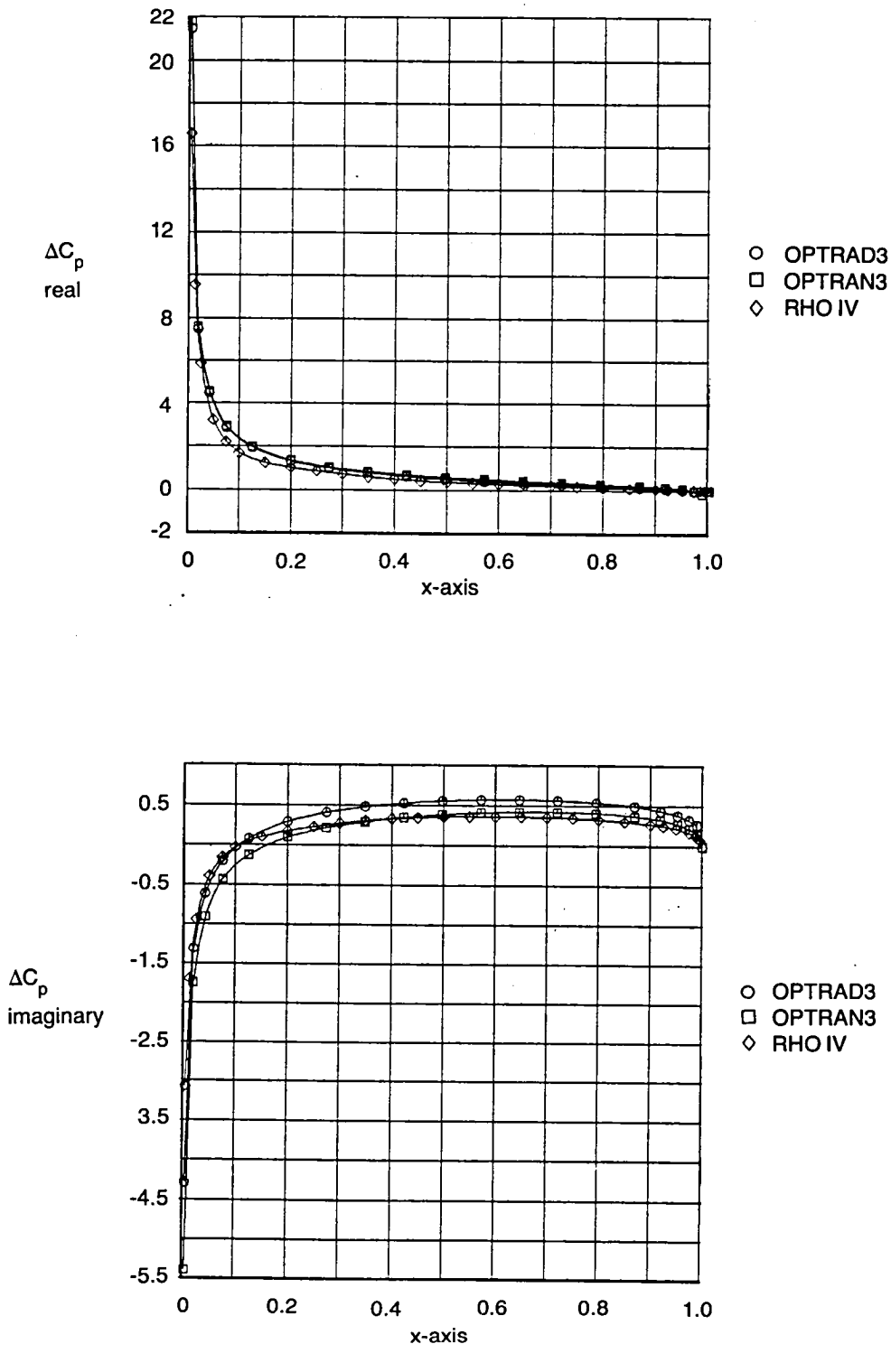
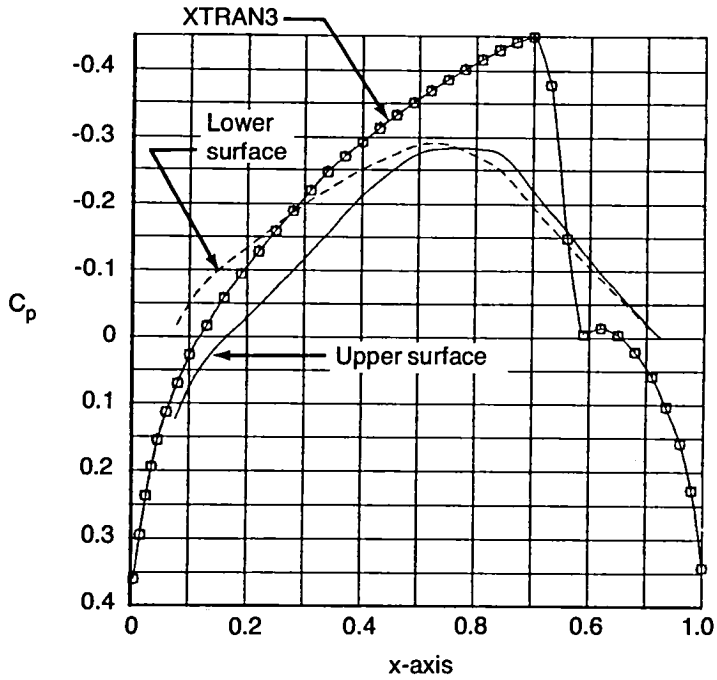
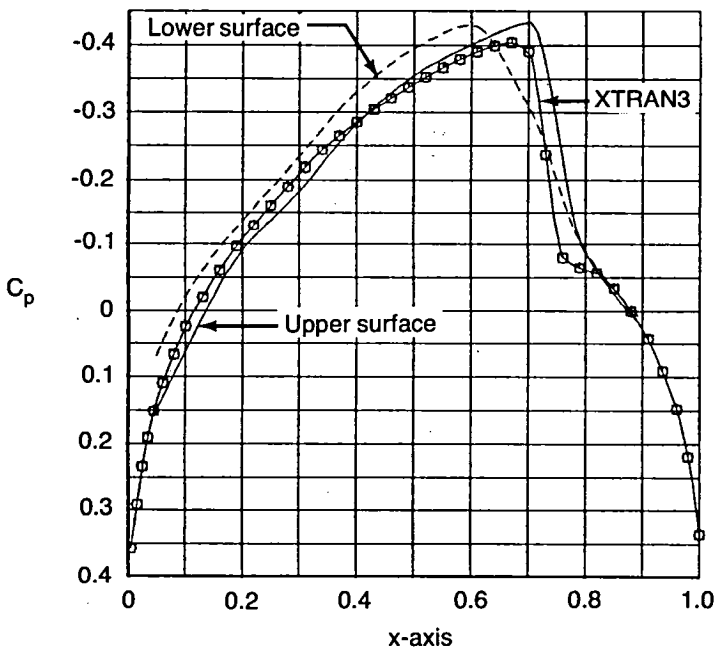


Figure 18.—Comparison of OPTRAD3 and OPTRAN3 With RHOIV, Pressure Distributions, Aspect Ratio 3 Rectangular Wing of Zero Thickness, $M = 0.9$, $k = 0.1$, $\bar{\eta} = 0.94$

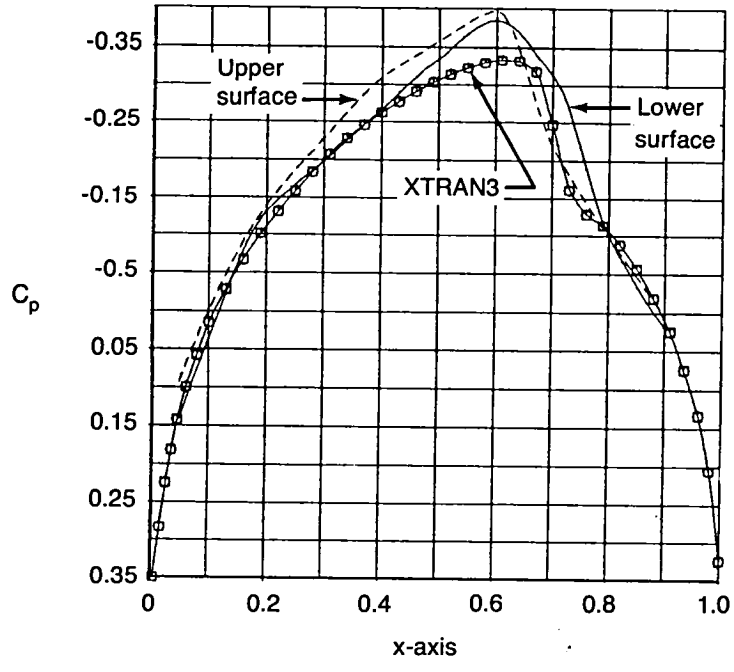


(a) Root chord

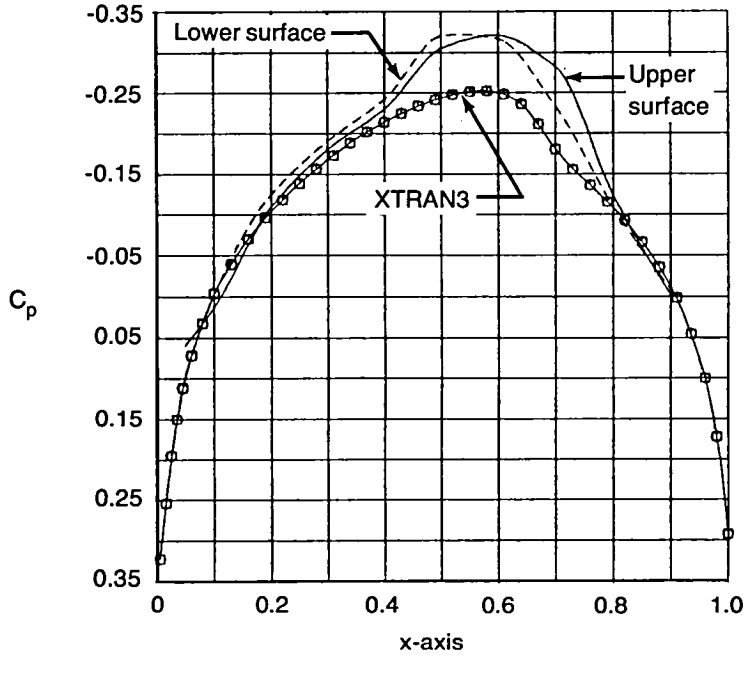


(b) Midspan

Figure 19.—Comparison of XTRAN3S With Experimental Result, Steady Pressure Distributions, Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil, M = 0.9



(c) 0.7 semispan



(d) 0.9 semispan

Figure 19.—Comparison of XTRAN3S With Experimental Result, Steady Pressure Distributions, Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil, M = 0.9 (Concluded)

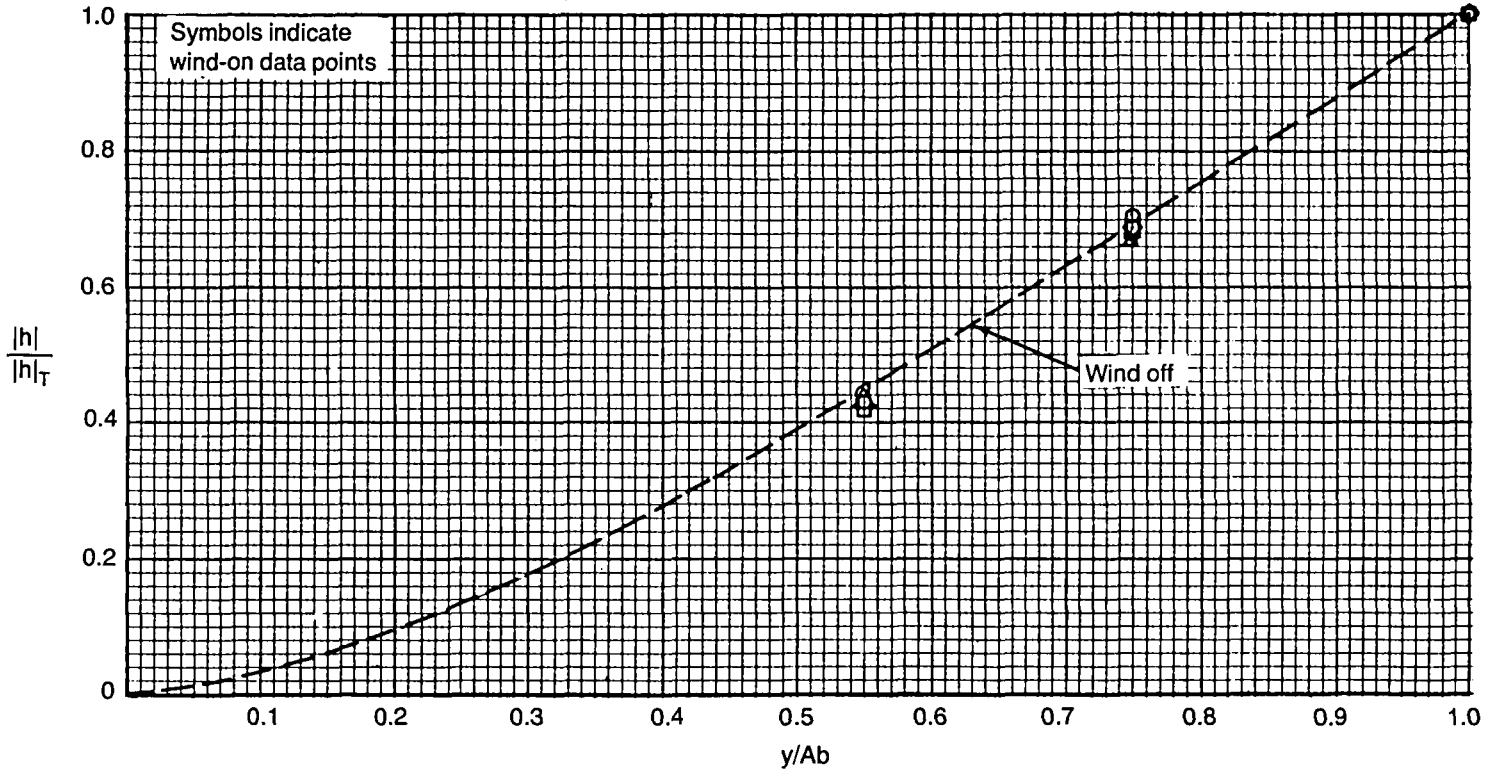
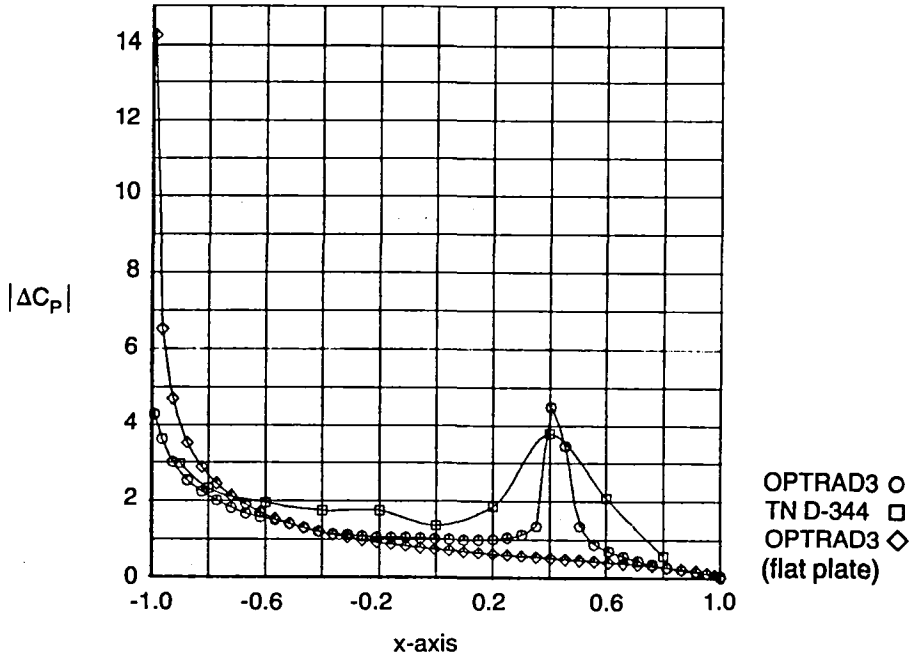
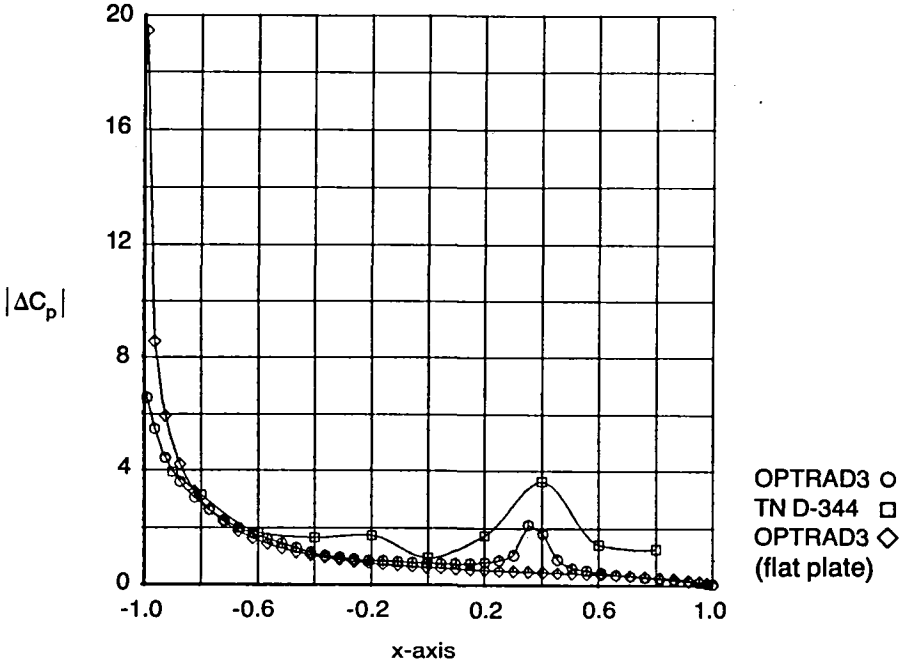


Figure 20.—The Normalized Bending Mode for the Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil

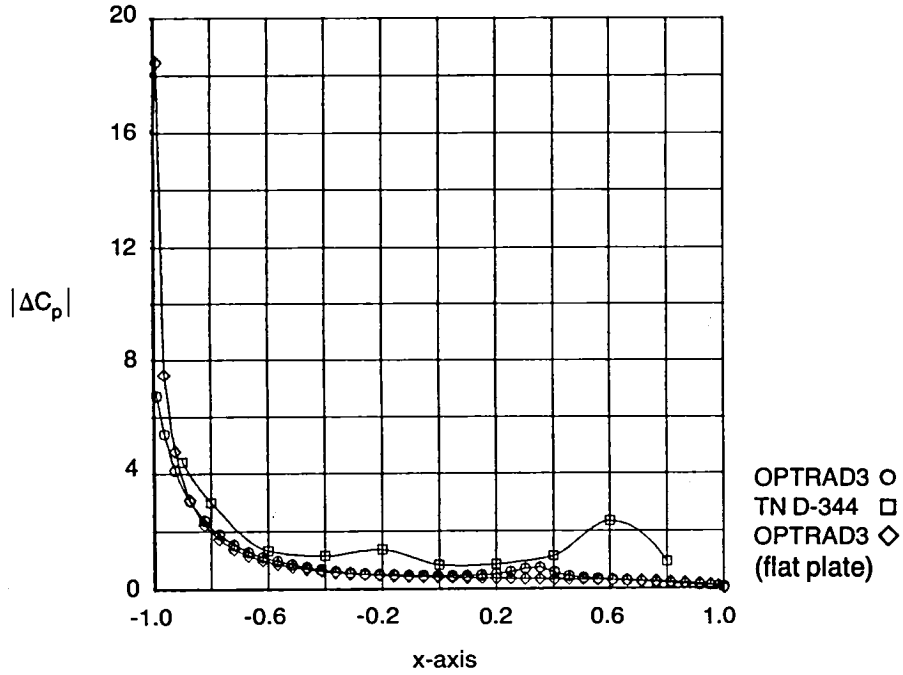


(a) Midsemispan



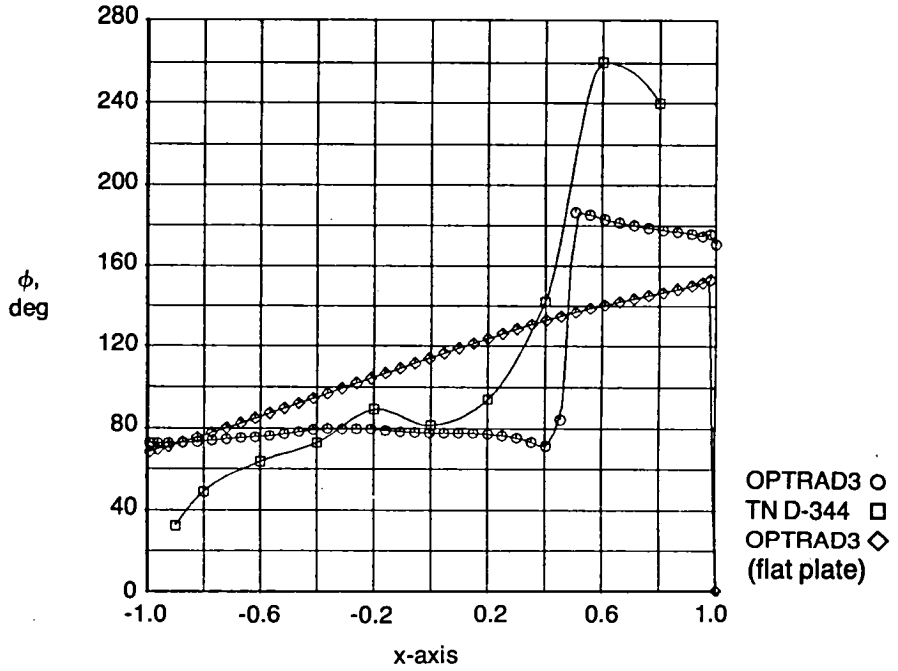
(b) 0.7 semispan

Figure 21.—Comparison of OPTRAD3 With Experimental Result, Pressure Amplitude Distributions, Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil, M = 0.9, k = 0.13

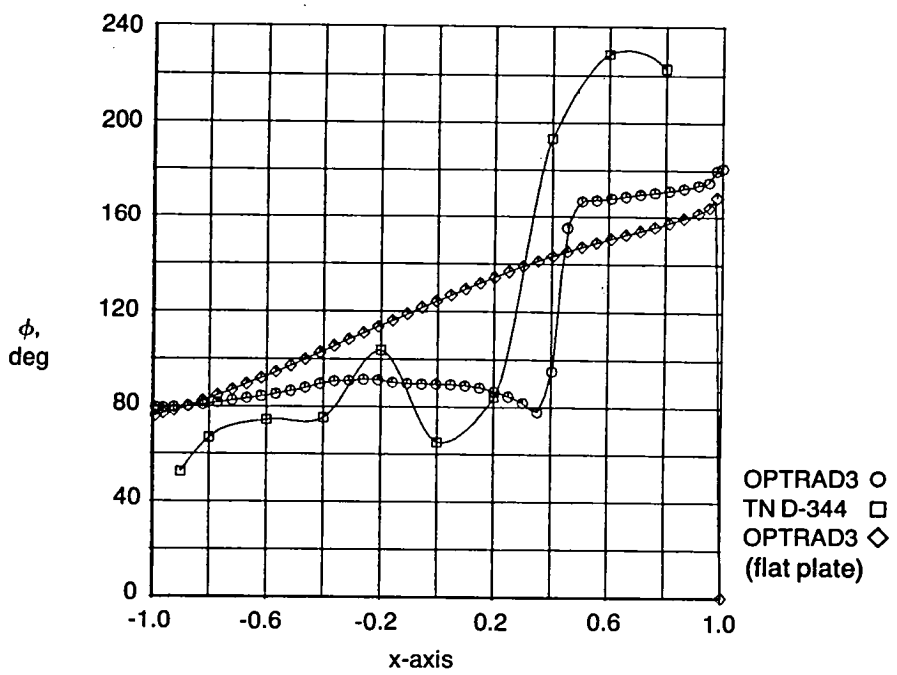


(c) 0.9 semispan

Figure 21.—Comparison of OPTRAD3 With Experimental Result, Pressure Amplitude Distributions, Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil, M = 0.9, k = 0.13 (Concluded)

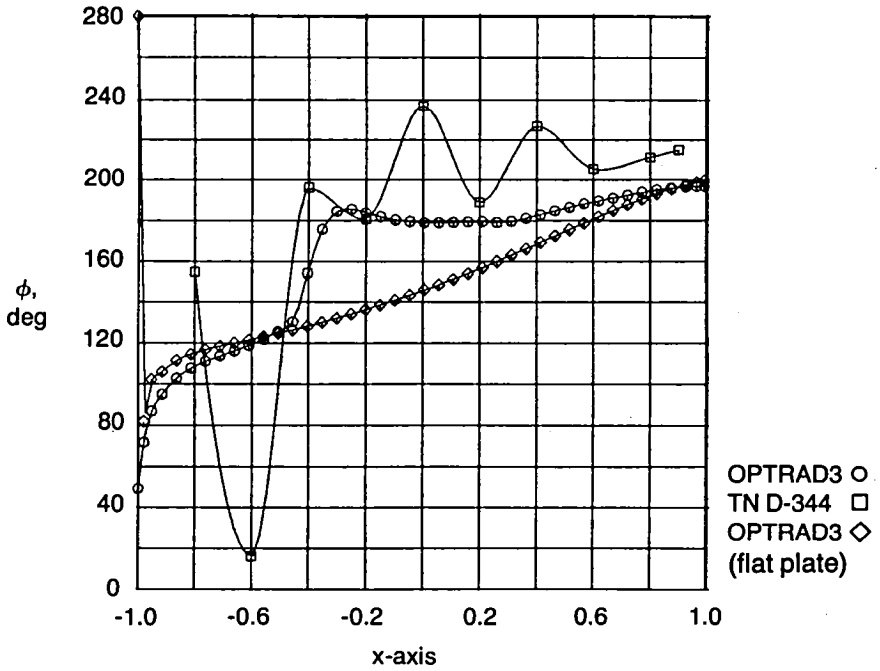


(a) Midsemispan



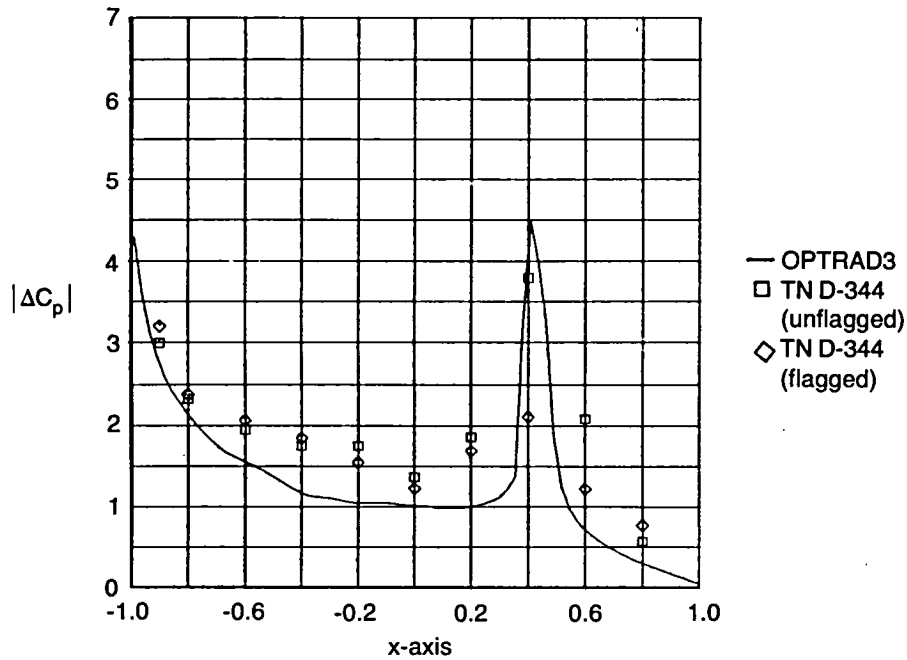
(b) 0.7 semispan

Figure 22.—Comparison of OPTRAD3 With Experimental Result, Pressure Phase-Angle Distributions, Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil, $M = 0.9$, $k = 0.13$

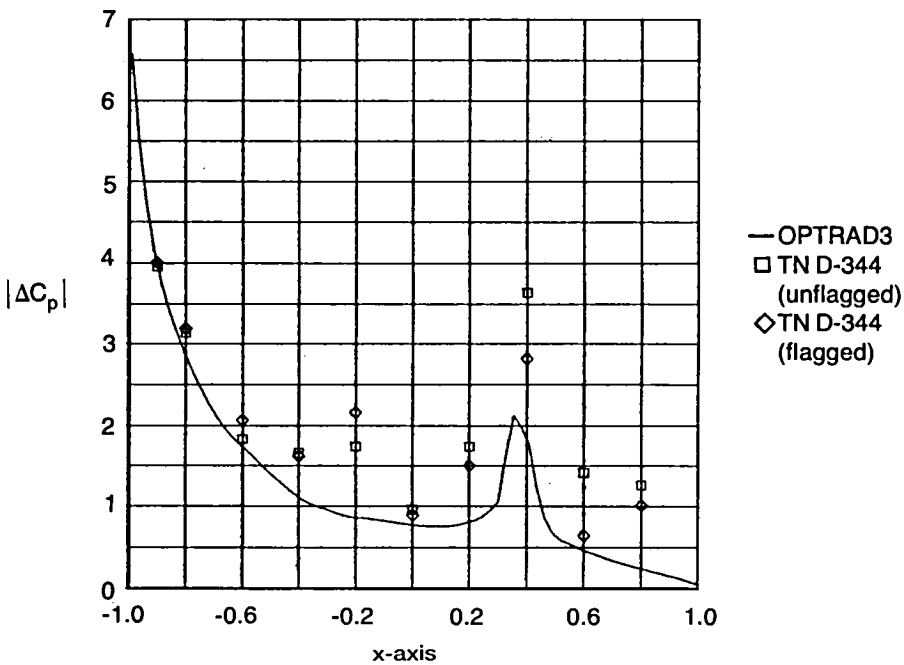


(c) 0.9 semispan

Figure 22.—Comparison of OPTRAD3 With Experimental Result, Pressure Phase-Angle Distributions, Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil, M = 0.9, k = 0.13 (Concluded)

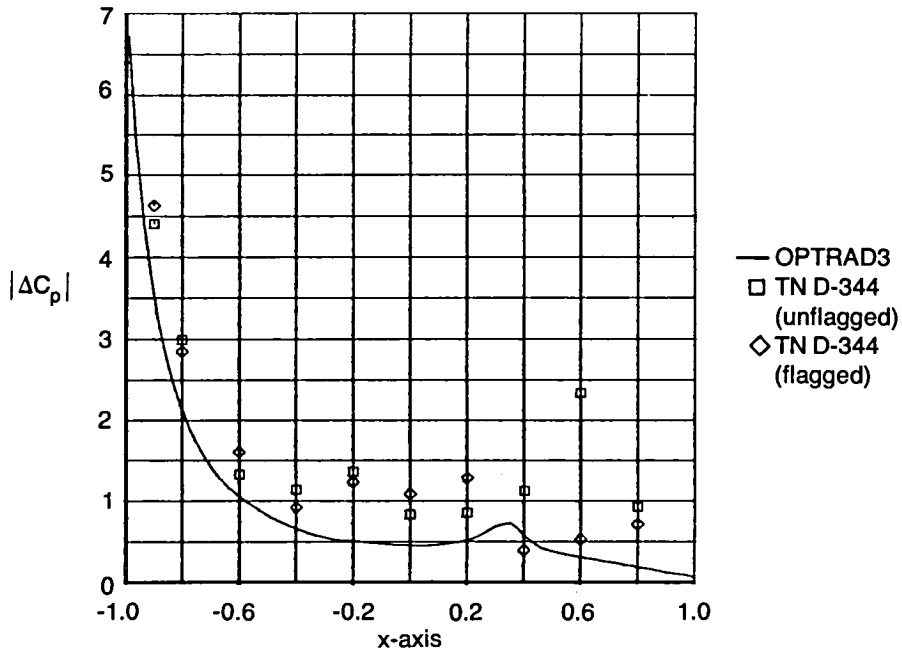


(a) Midsemispan



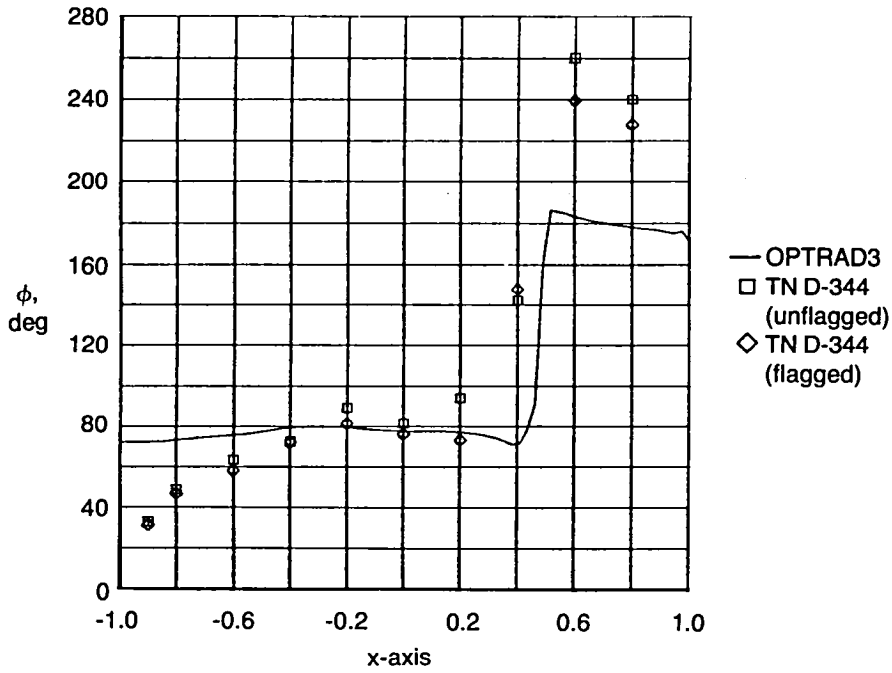
(b) 0.7 semispan

Figure 23.—Comparison of OPTRAD3 With Two Experimental Results, Pressure Amplitude Distributions, Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil, $M = 0.9, k = 0.13$

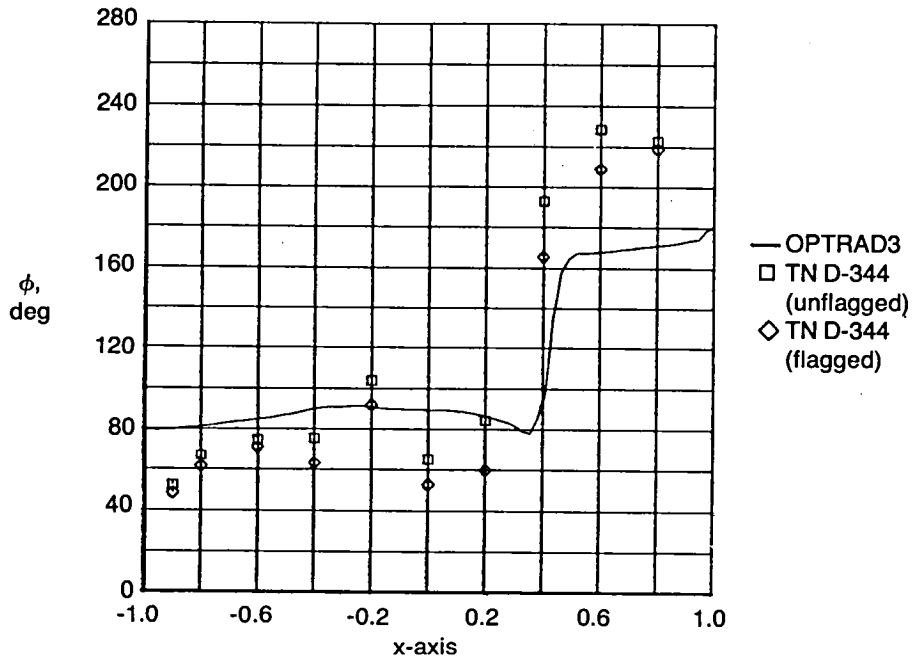


(c) 0.9 semispan

Figure 23.—Comparison of OPTRAD3 With Two Experimental Results, Pressure Amplitude Distributions, Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil, M = 0.9, k = 0.13 (Concluded)

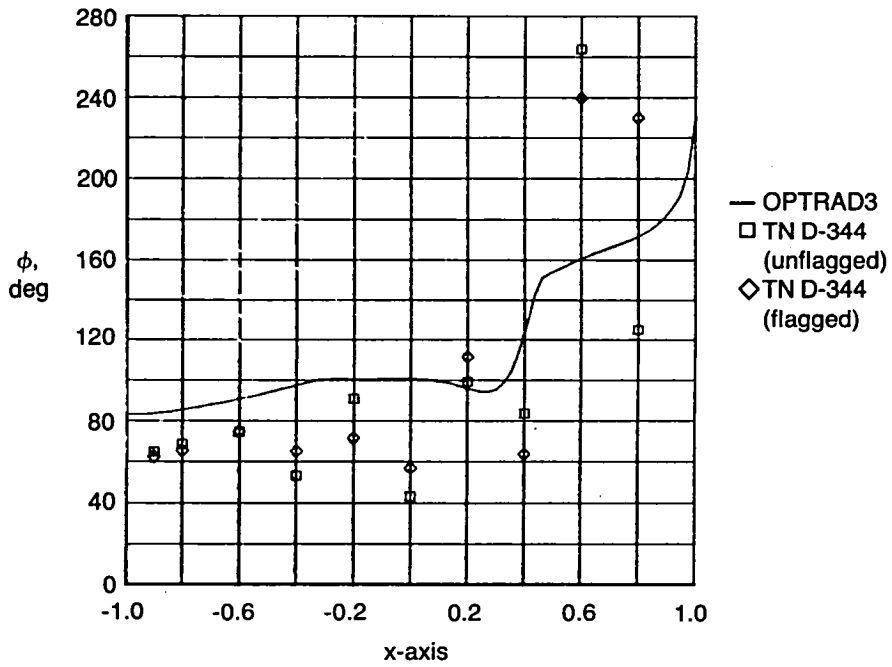


(a) Midsemispan



(b) 0.7 semispan

Figure 24.—Comparison of OPTRAD3 With Two Experimental Results, Pressure Phase-Angle Distributions, Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil, M = 0.9, k = 0.13



(c) 0.9 semispan

Figure 24.—Comparison of OPTRAD3 With Two Experimental Results, Pressure Phase-Angle Distributions, Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil, M = 0.9, k = 0.13 (Concluded)

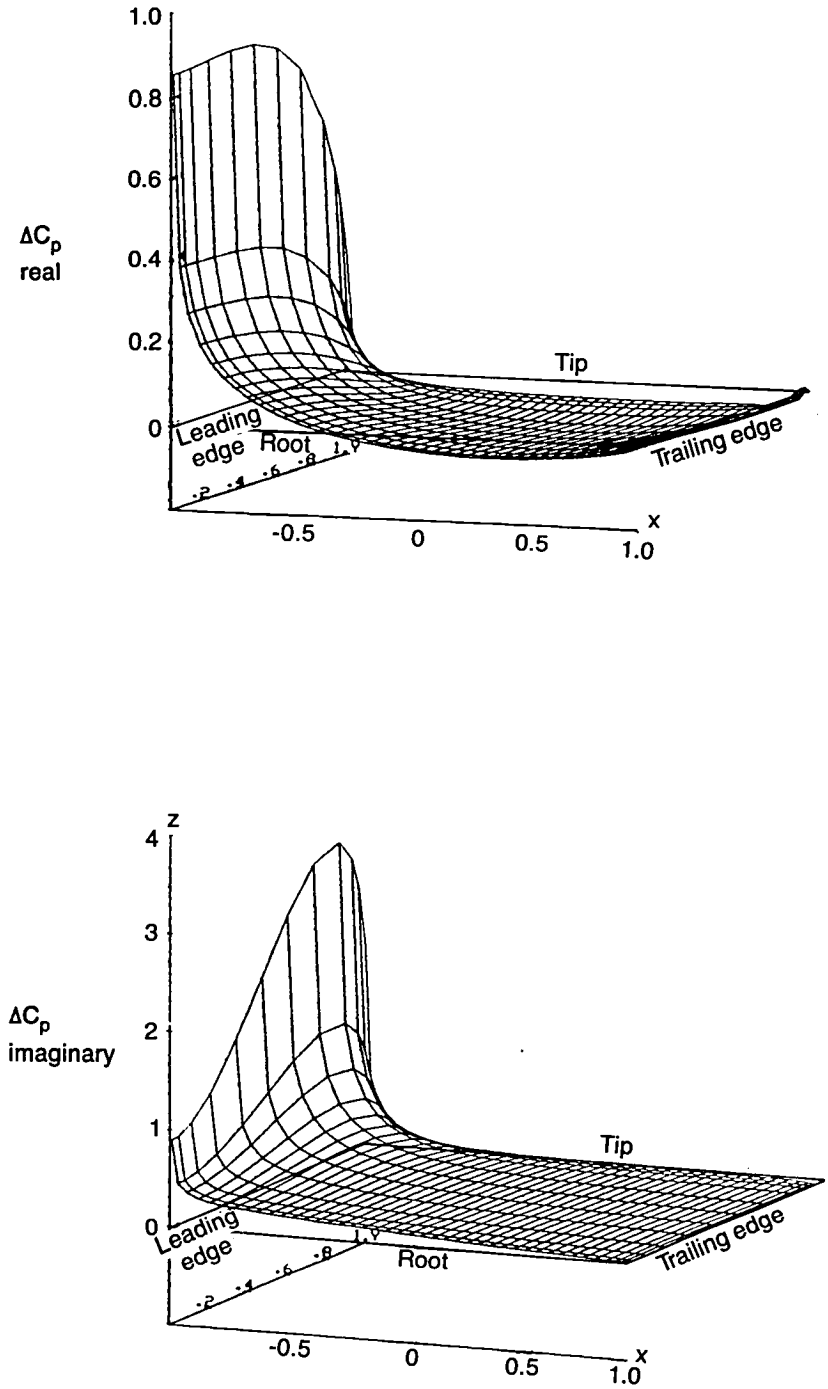


Figure 25.—Three-Dimensional Representation of the Pressure Distribution for an Aspect Ratio 3 Rectangular Wing With Zero Thickness Airfoil, M = 0.9, k = 0.13

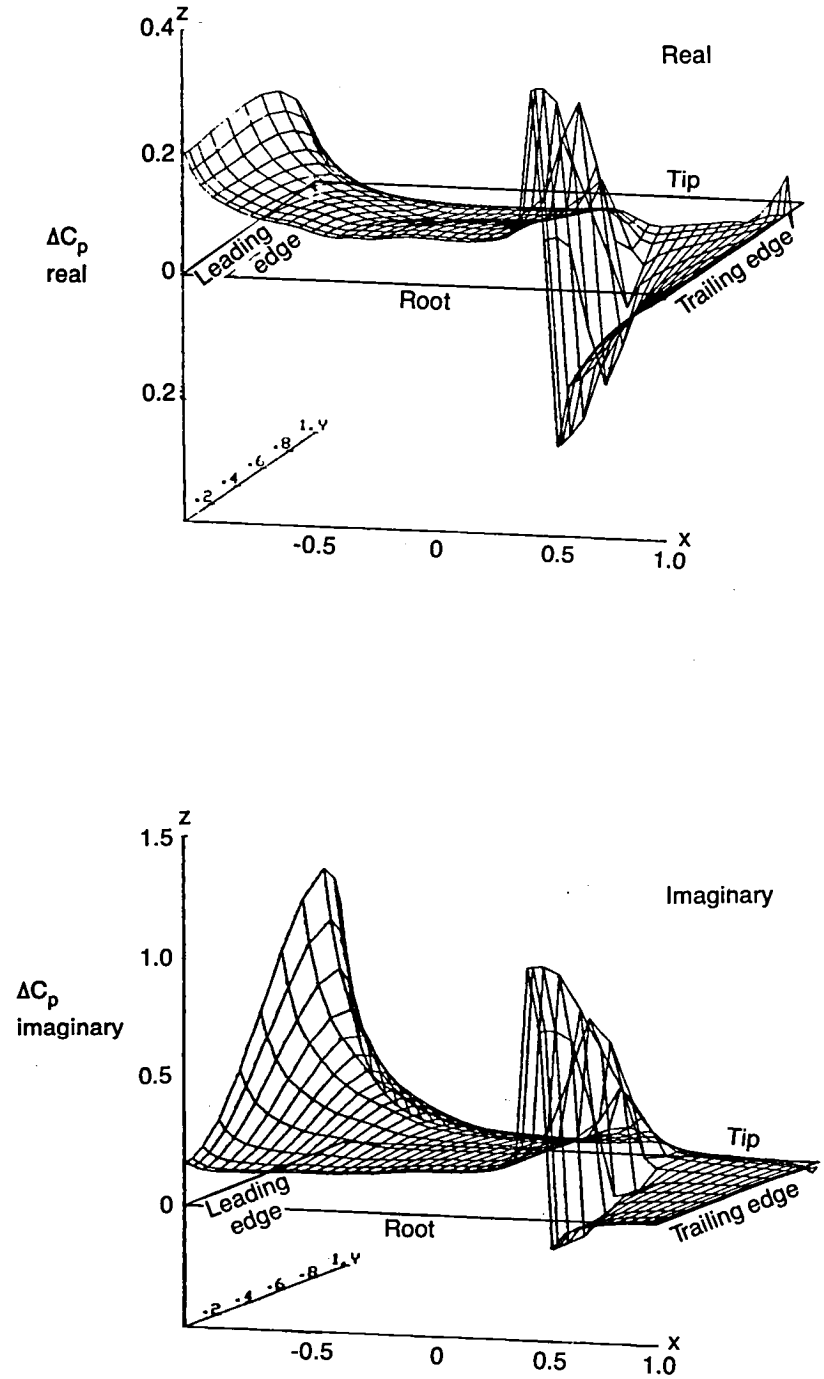


Figure 26.—Three-Dimensional Representation of the Pressure Distribution for an Aspect Ratio 3 Rectangular Wing With Circular Arc Airfoil, M = 0.9, k = 0.13

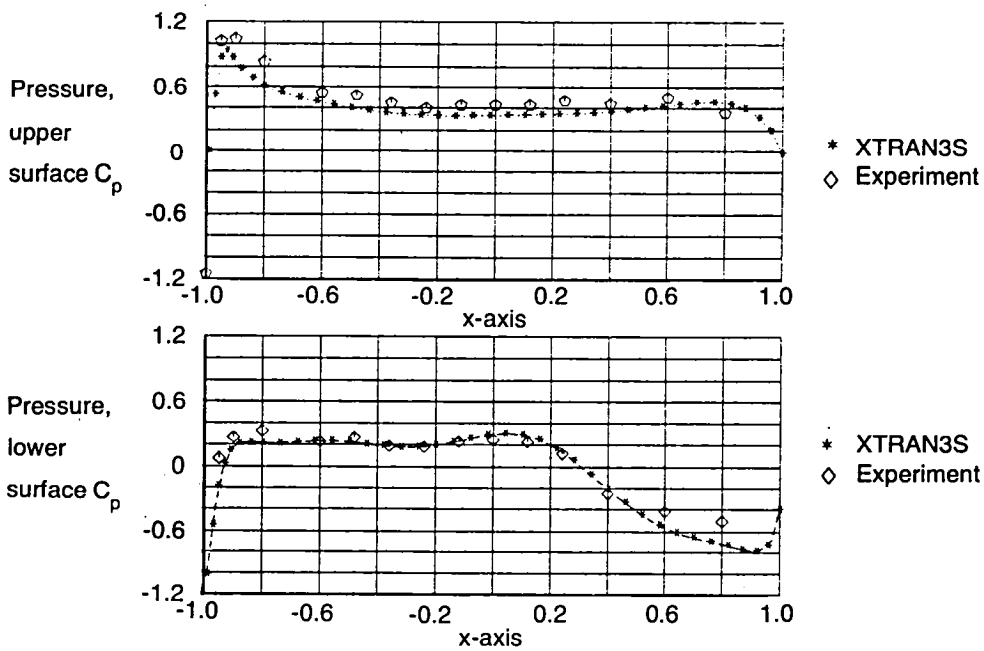


Figure 27.—Correlation of XTRAN3S With Experimental Result, Steady Pressure Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, $M = 0.7$, $\bar{\eta} = 0.31$

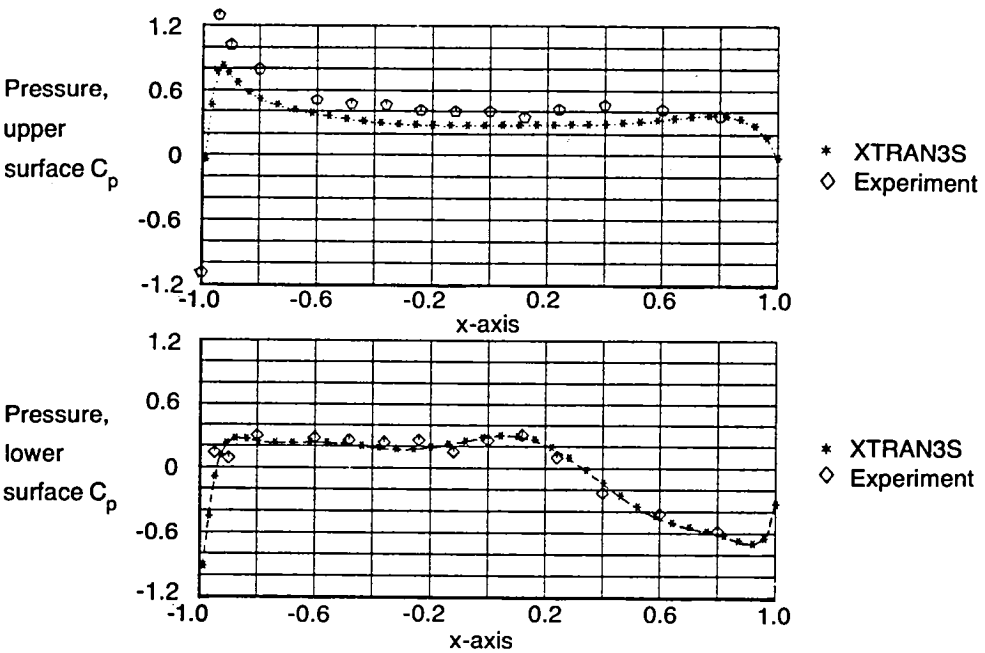


Figure 28.—Correlation of XTRAN3S With Experimental Result, Steady Pressure Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, $M = 0.7$, $\bar{\eta} = 0.59$

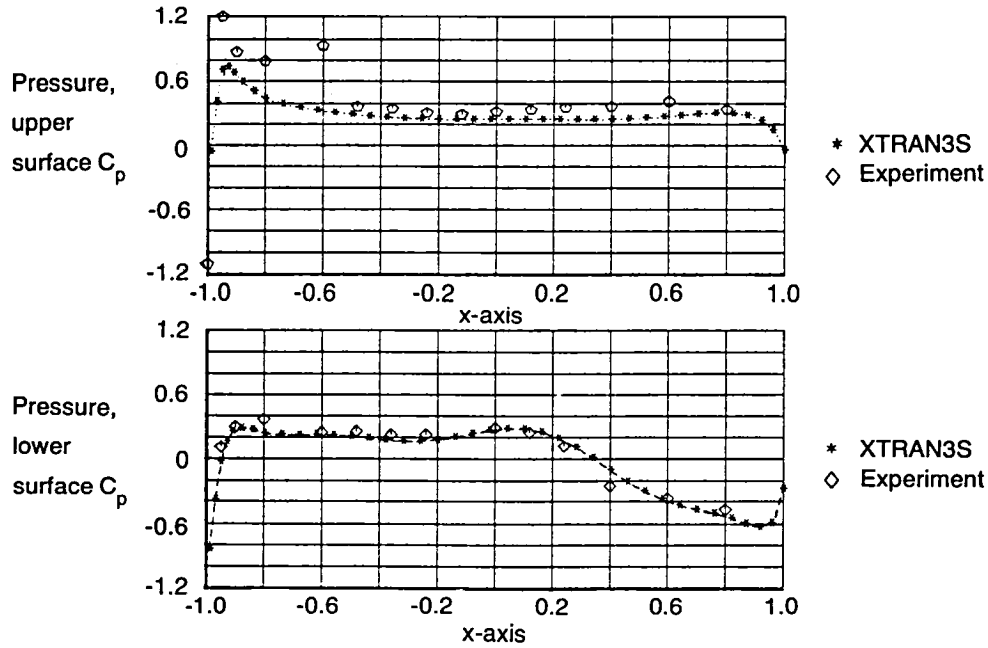


Figure 29.—Correlation of XTRAN3S With Experimental Result, Steady Pressure Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, $M = 0.7$, $\bar{\eta} = 0.81$

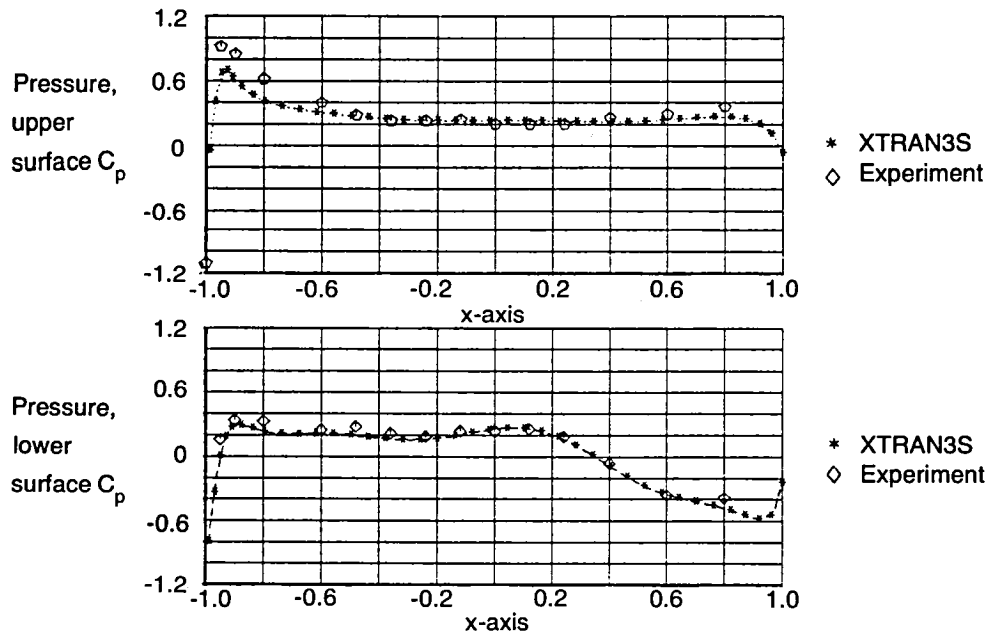
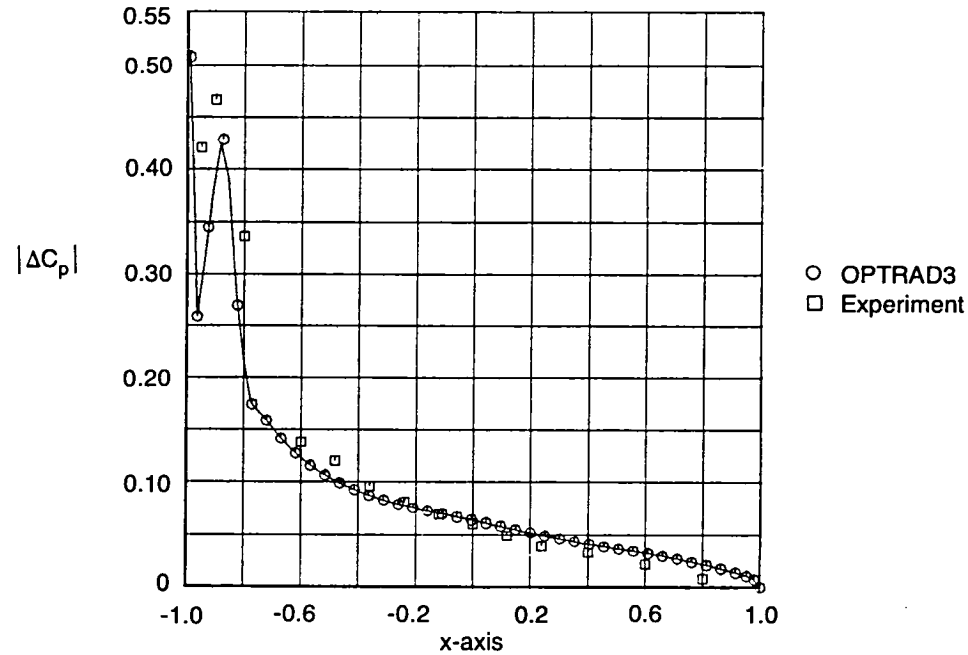
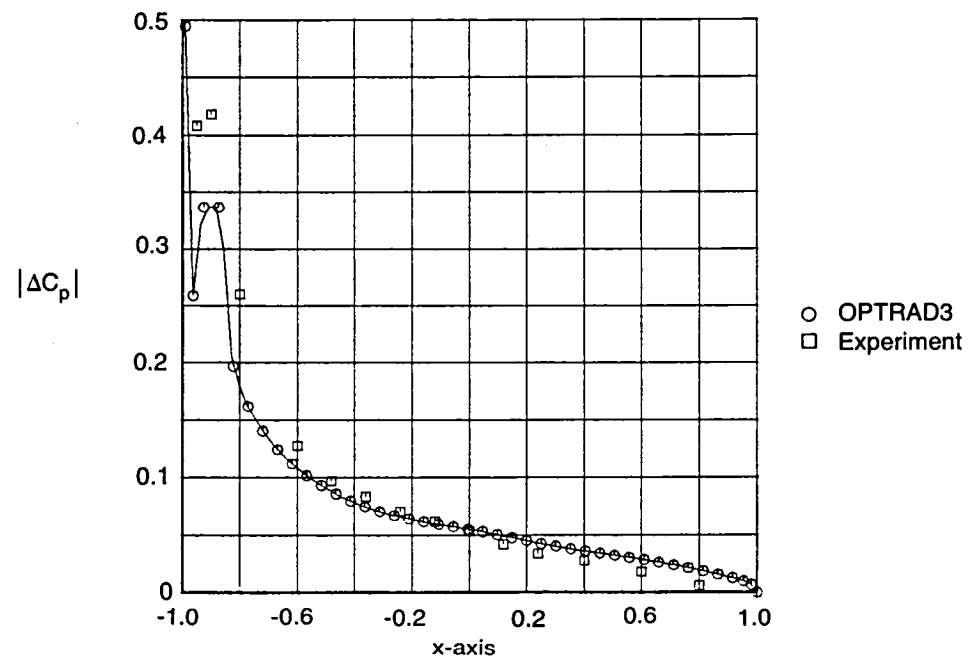


Figure 30.—Correlation of XTRAN3S With Experimental Result, Steady Pressure Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, $M = 0.7$, $\bar{\eta} = 0.95$

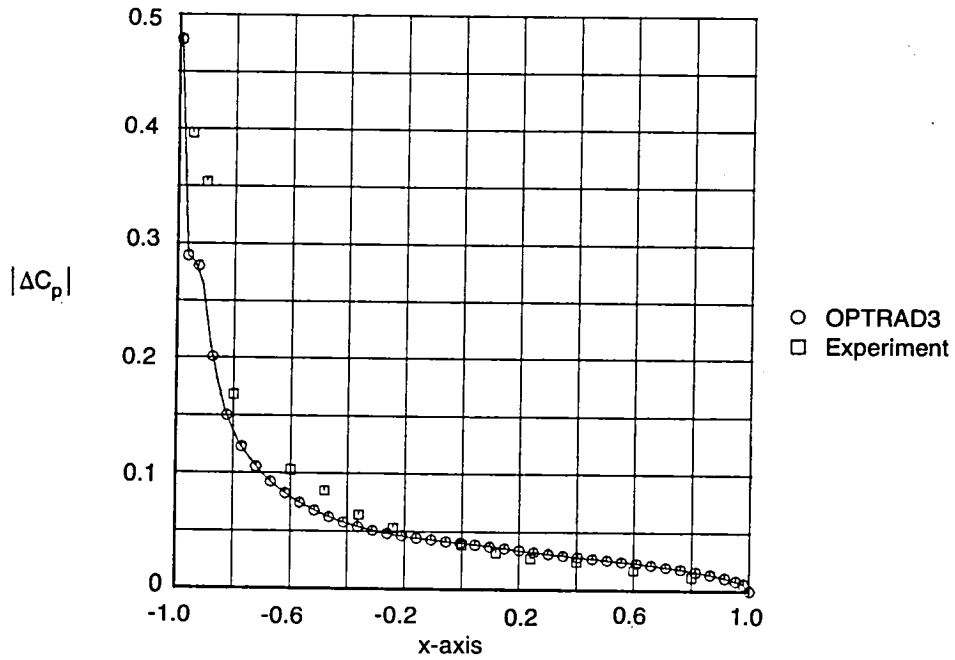


(a) 0.31 semispan

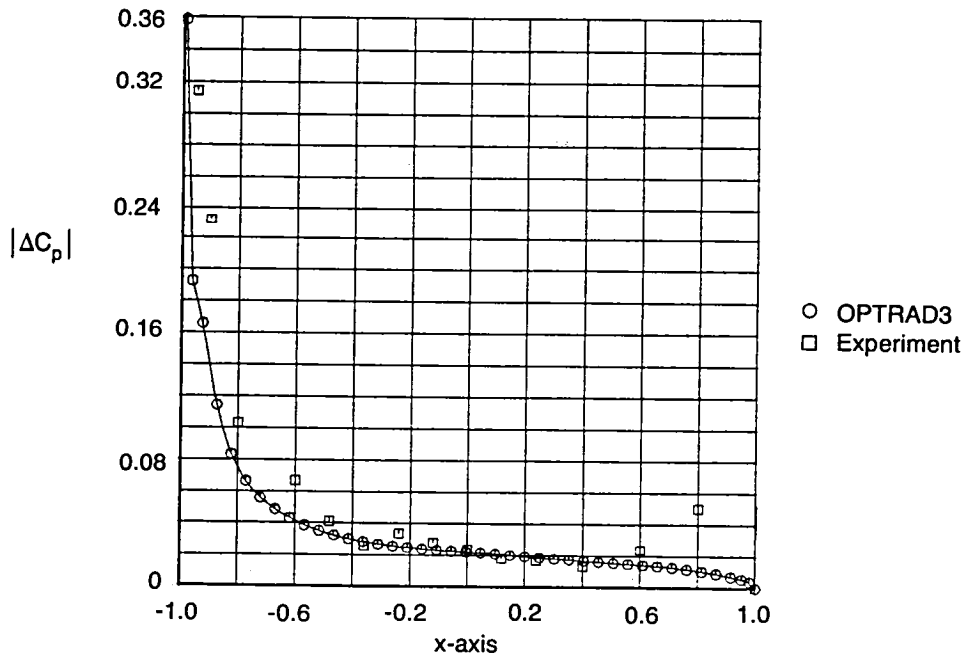


(b) 0.59 semispan

Figure 31.—Correlation of OPTRAD3 With Experimental Result, Pressure Amplitude Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.178

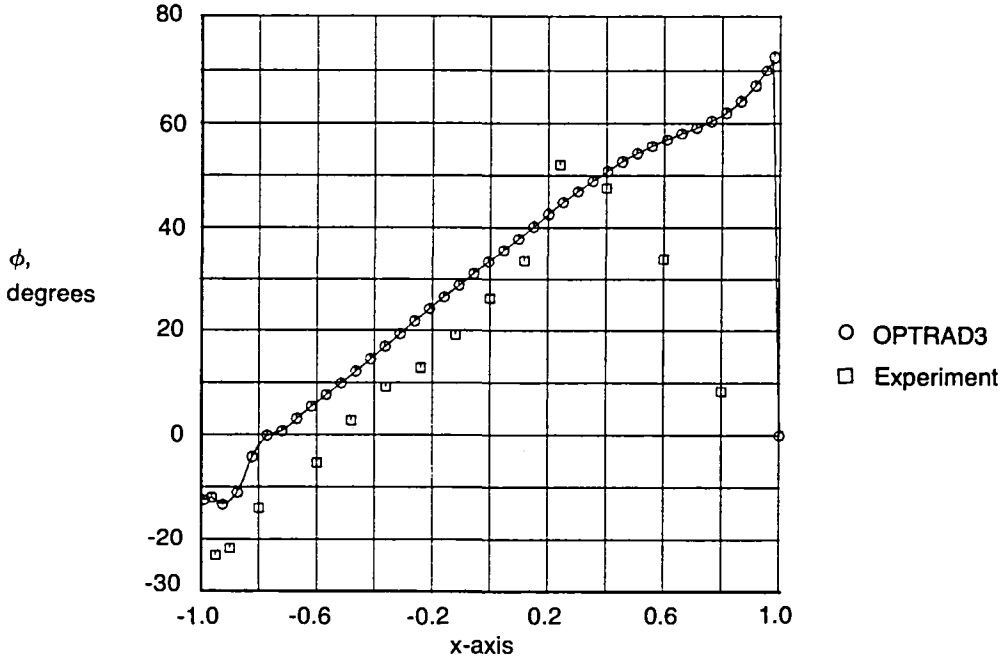


(c) 0.81 semispan

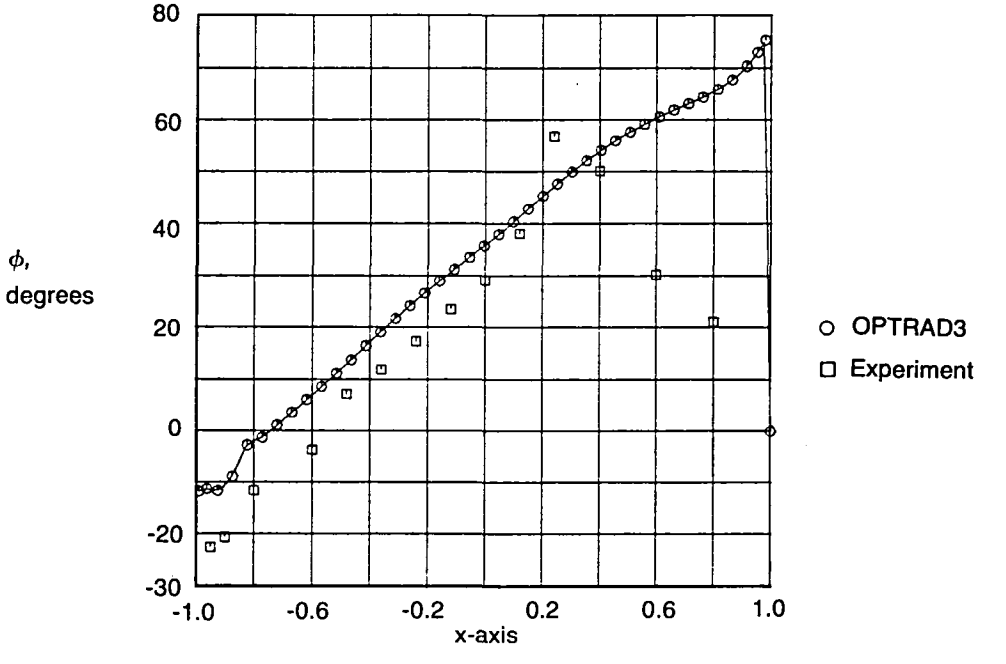


(d) 0.95 semispan

Figure 31.—Correlation of OPTRAD3 With Experimental Result, Pressure Amplitude Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.178 (Concluded)

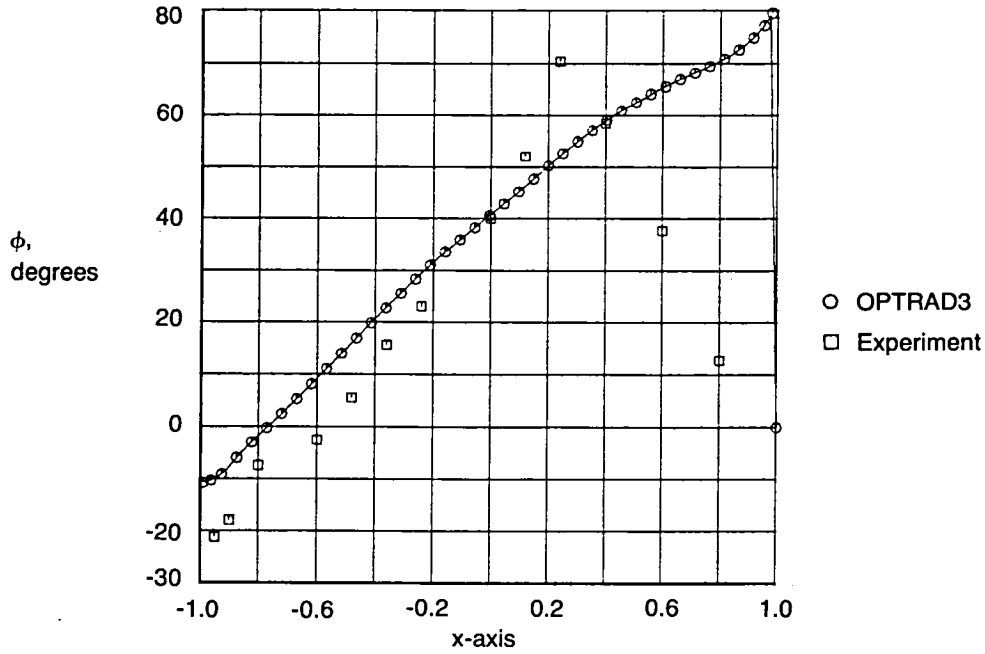


(a) 0.31 semispan

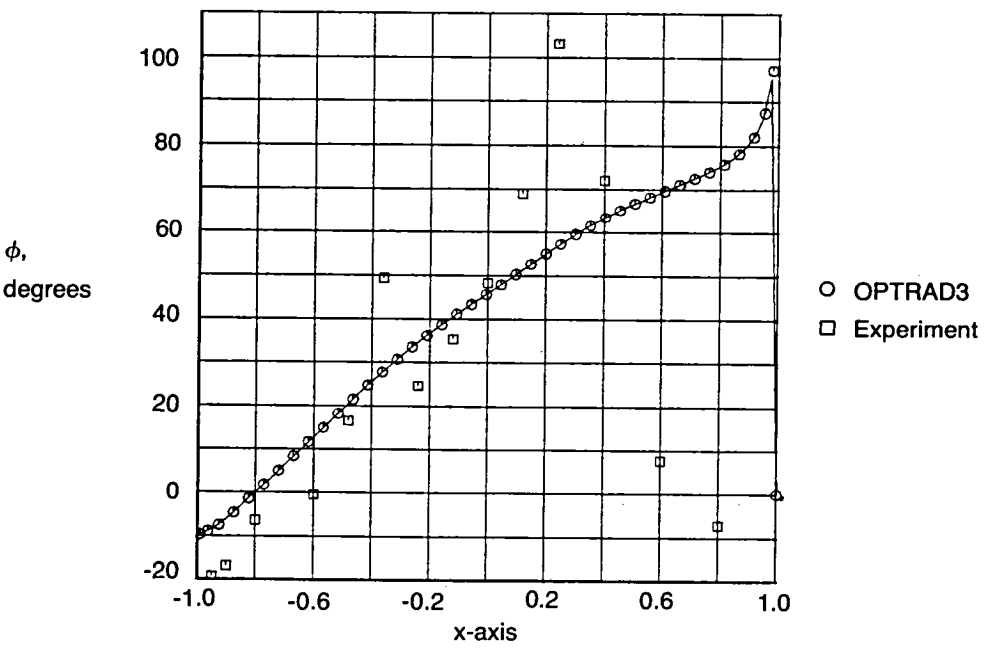


(b) 0.59 semispan

Figure 32.—Comparison of OPTRAD3 With Experimental Result, Pressure Phase-Angle Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, $M = 0.7$, $k = 0.178$

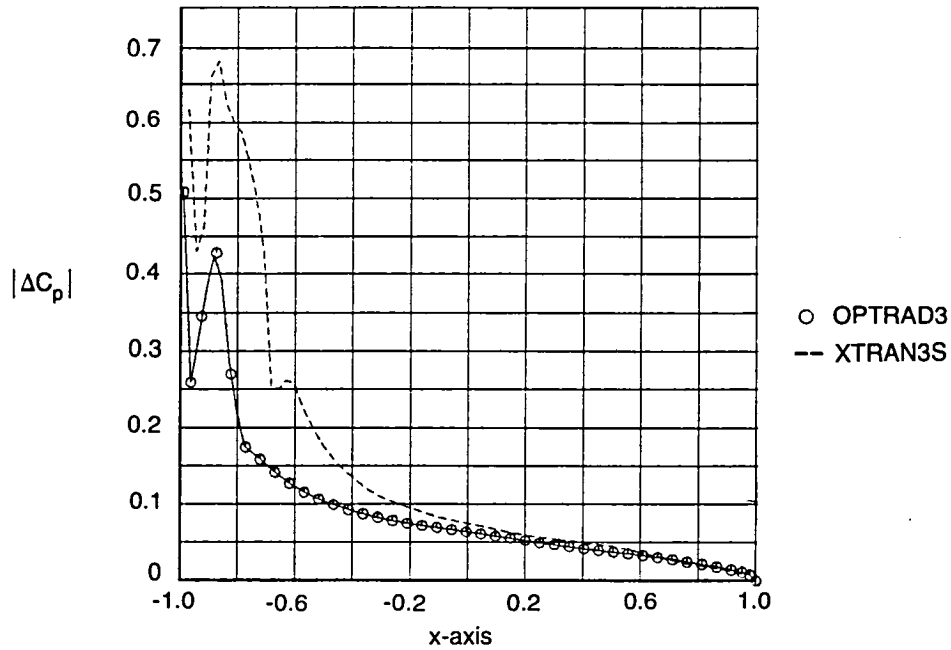


(c) 0.81 semispan

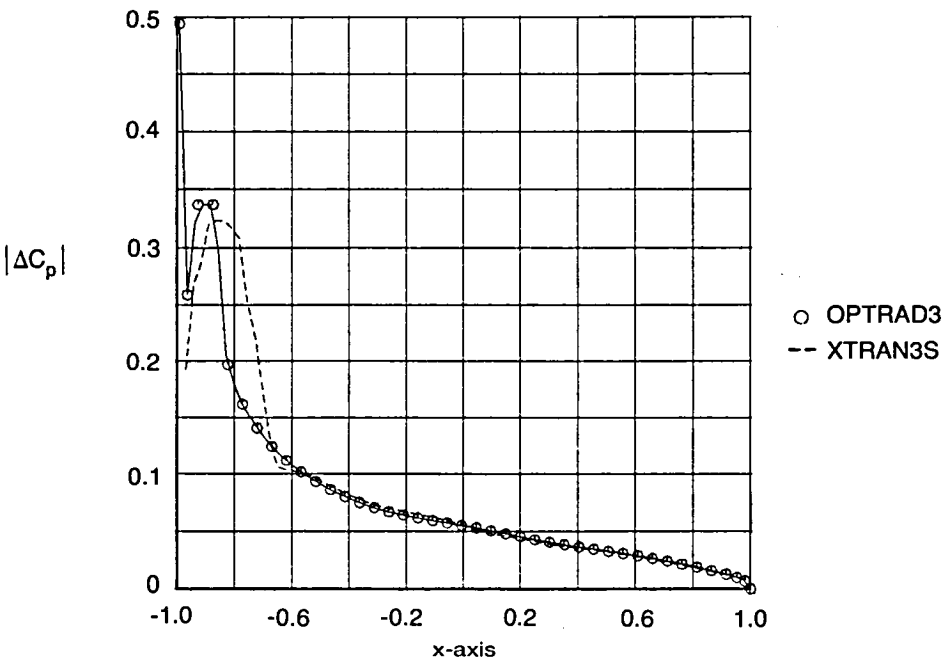


(d) 0.95 semispan

Figure 32.—Comparison of OPTRAD3 With Experimental Result, Pressure Phase-Angle Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.178 (Concluded)

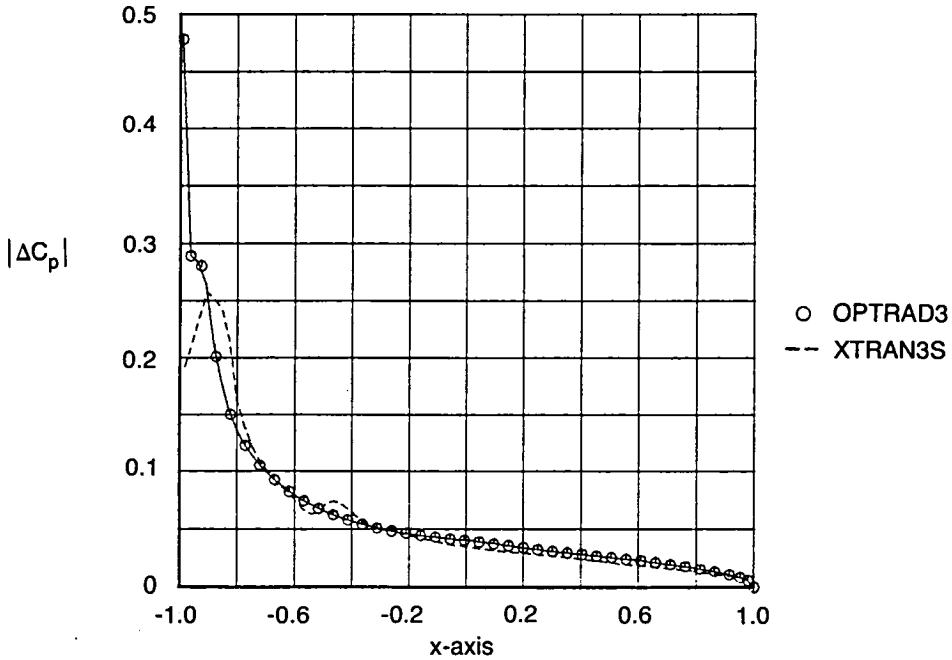


(a) 0.31 semispan

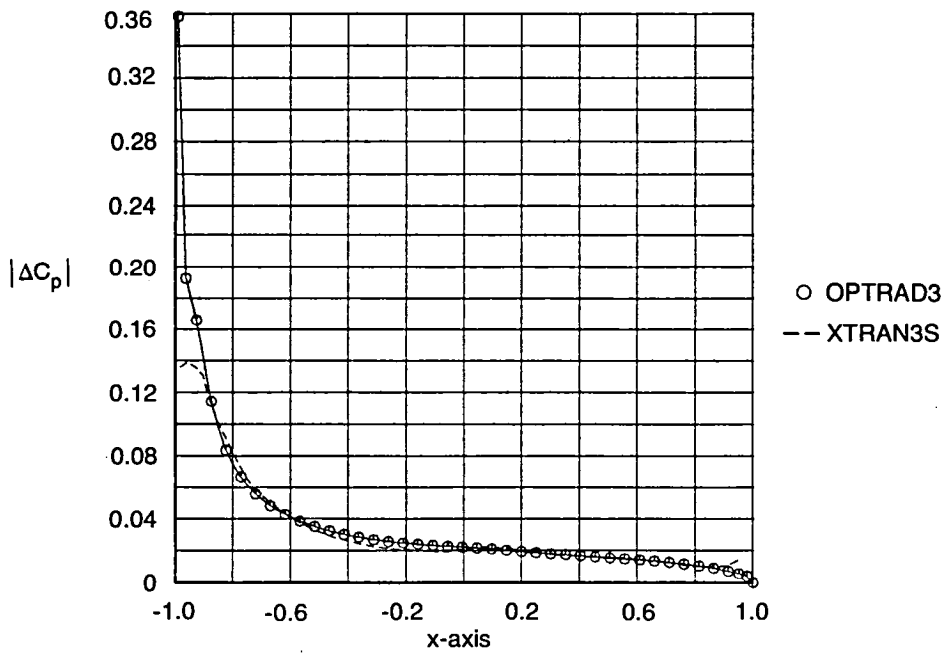


(b) 0.59 semispan

Figure 33.—Comparison of OPTRAD3 With XTRAN3S, Pressure Amplitude Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, $M = 0.7$, $k = 0.178$

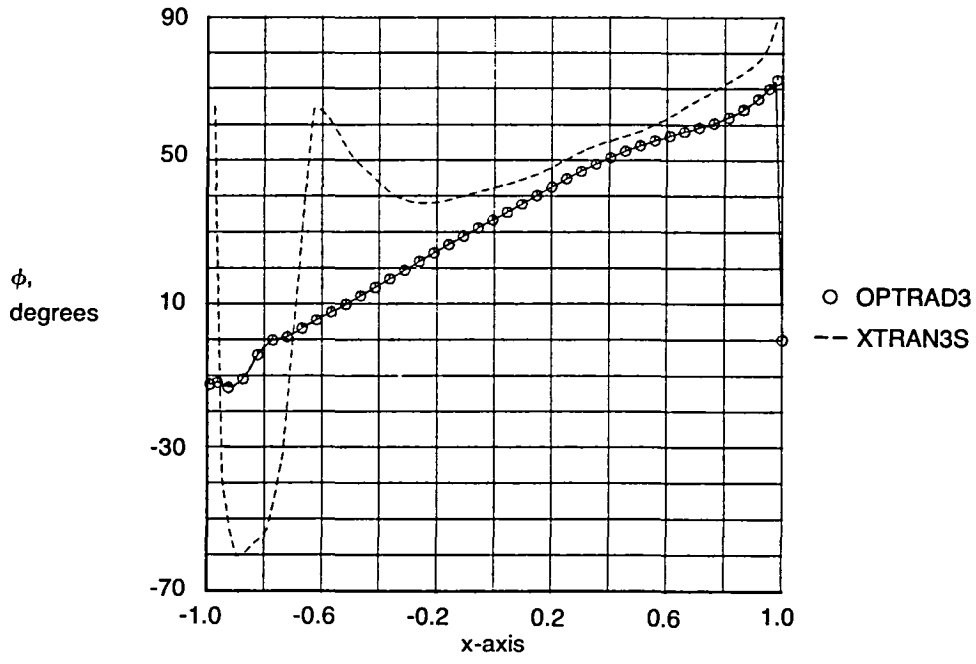


(c) 0.81 semispan

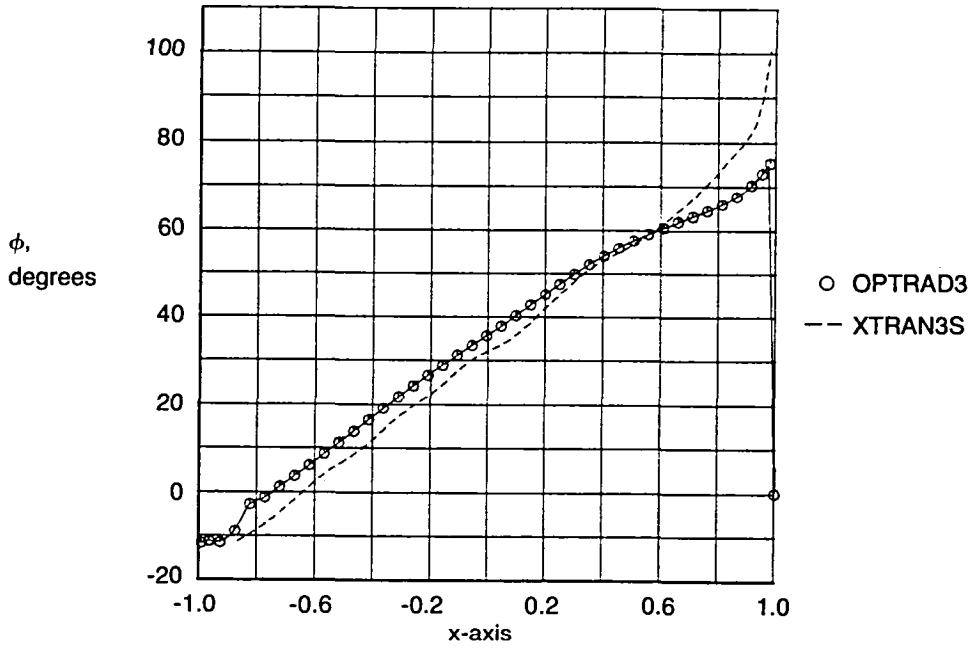


(d) 0.95 semispan

Figure 33.—Comparison of OPTRAD3 With XTRAN3S, Pressure Amplitude Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.178 (Concluded)

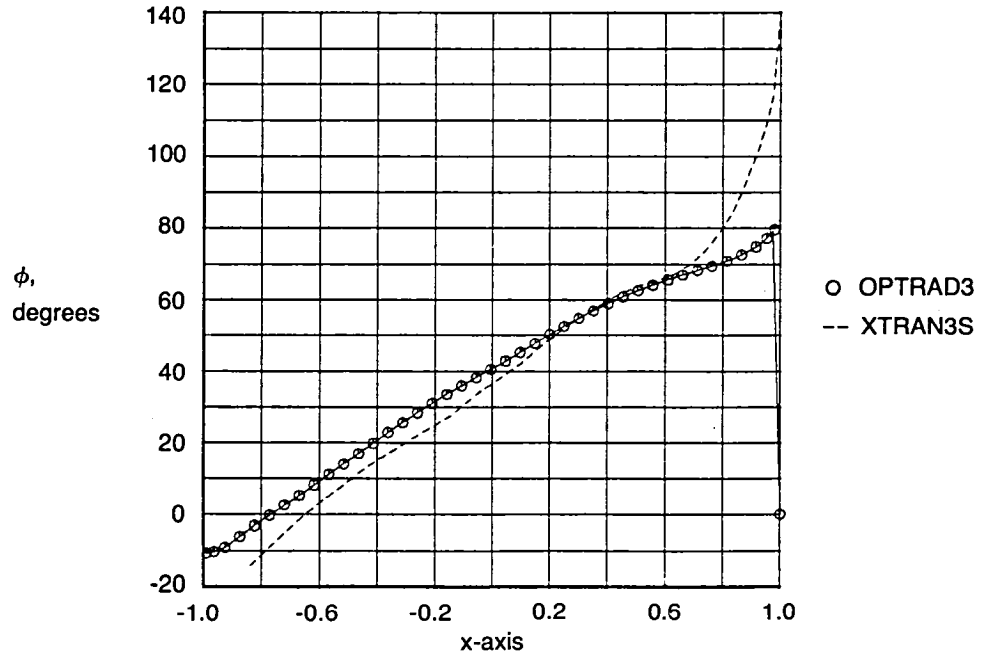


(a) 0.31 semispan

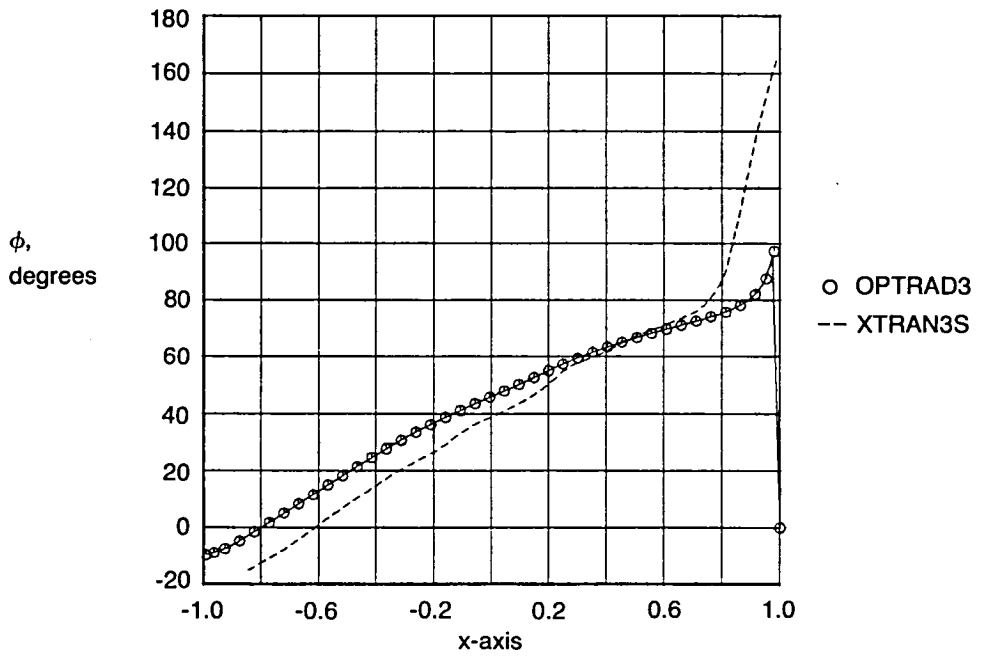


(b) 0.59 semispan

Figure 34.—Comparison of OPTRAD3 With XTRAN3S, Pressure Phase-Angle Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, $M = 0.7$, $k = 0.178$

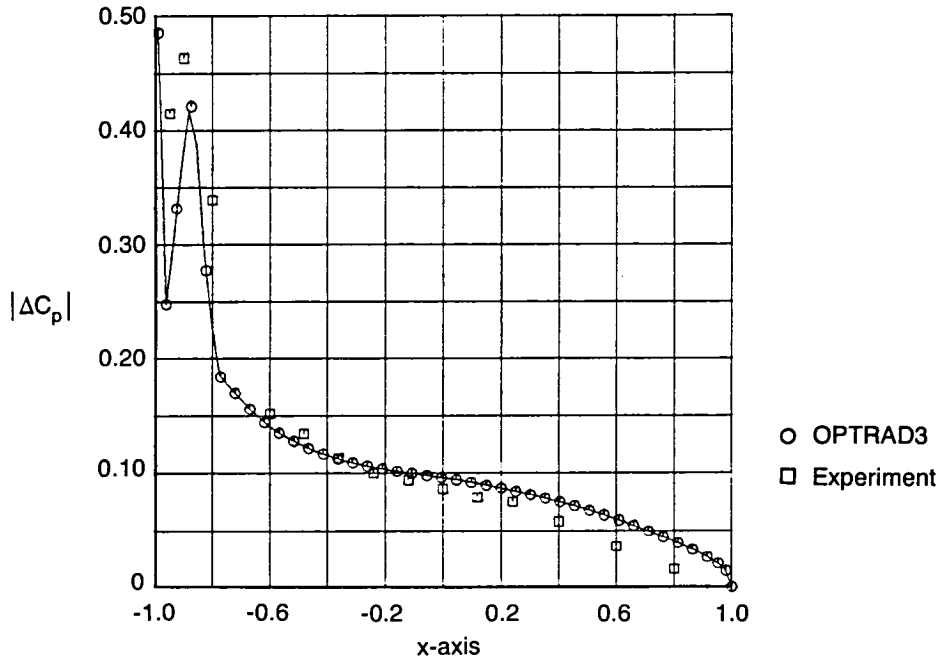


(c) 0.81 semispan

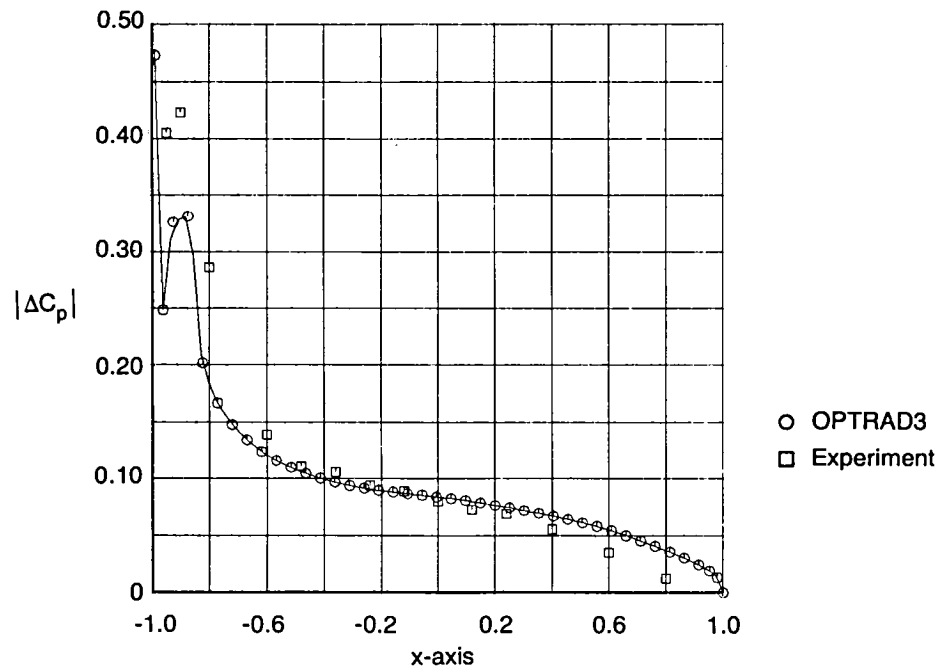


(d) 0.95 semispan

Figure 34.—Comparison of OPTRAD3 With XTRAN3S, Pressure Phase-Angle Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.178 (Concluded)

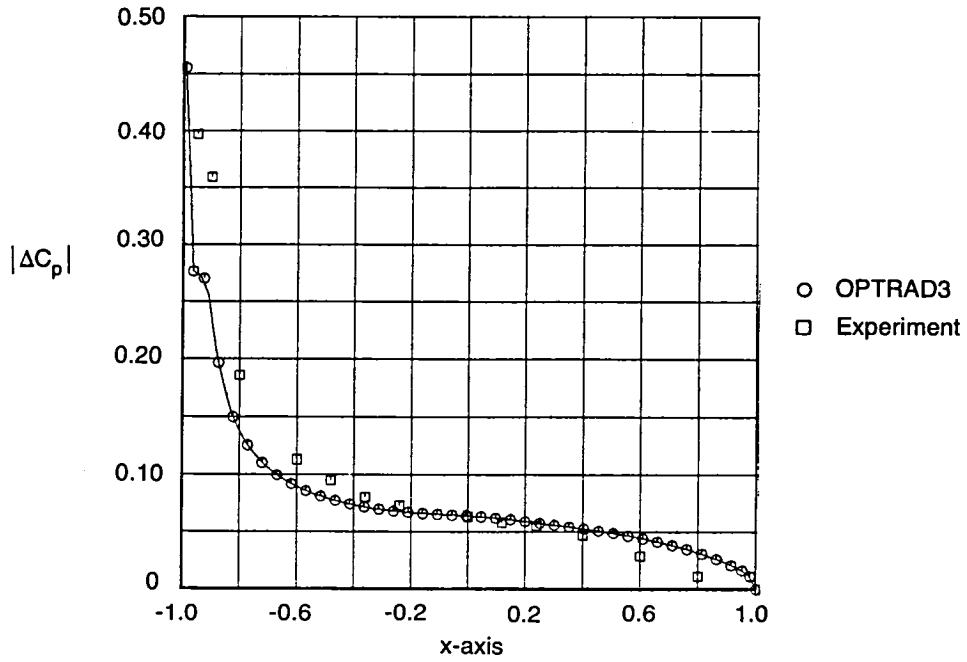


(a) 0.31 semispan

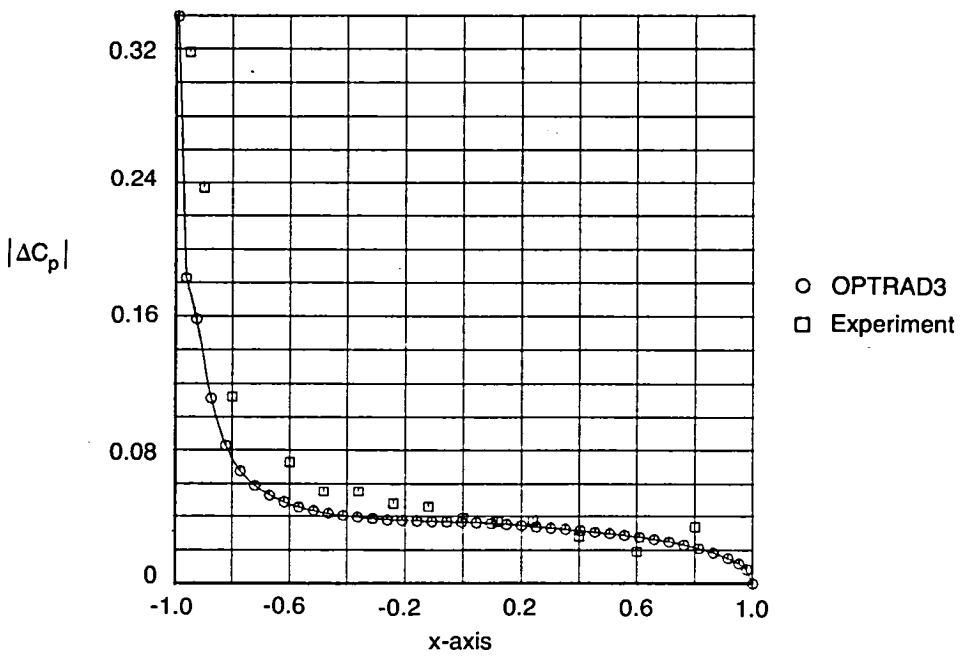


(b) 0.59 semispan

Figure 35.—Correlation of OPTRAD3 With Experimental Result, Pressure Amplitude Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.356

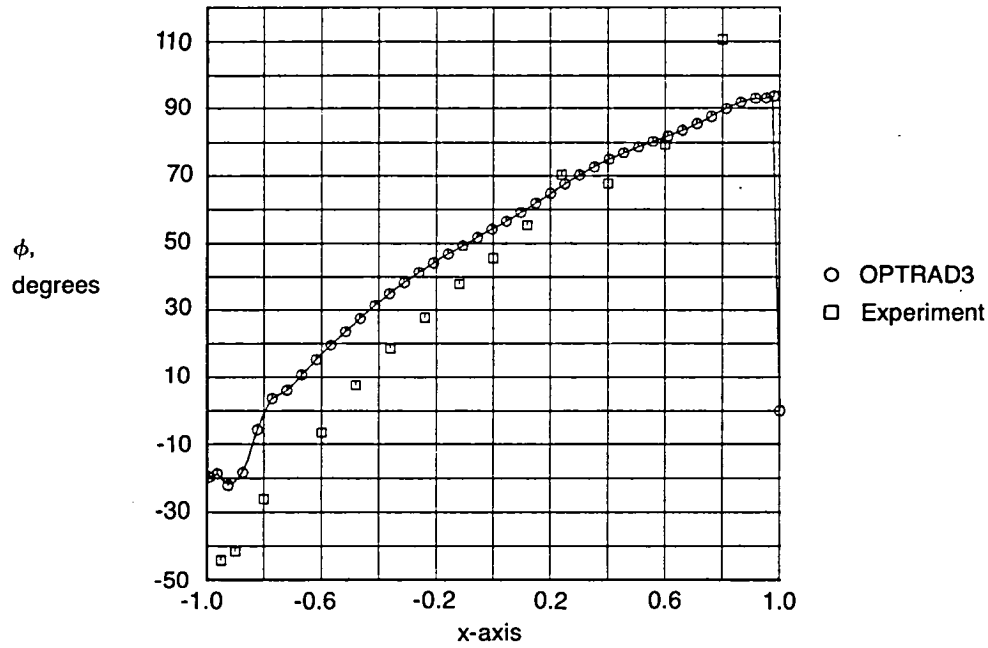


(c) 0.81 semispan

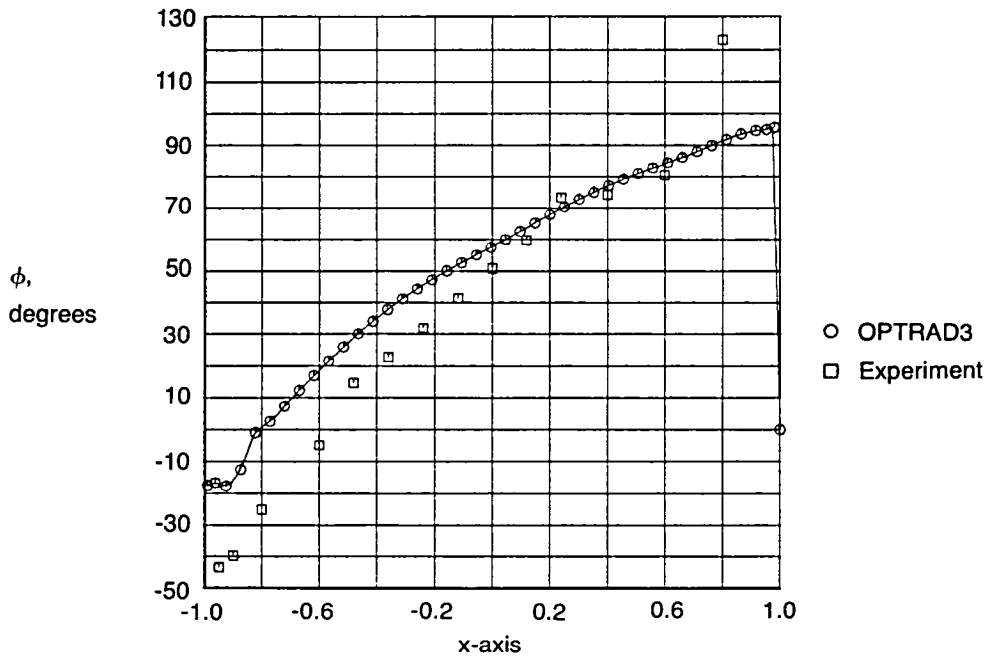


(d) 0.95 semispan

Figure 35.—Correlation of OPTRAD3 With Experimental Result, Pressure Amplitude Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.356 (Concluded)

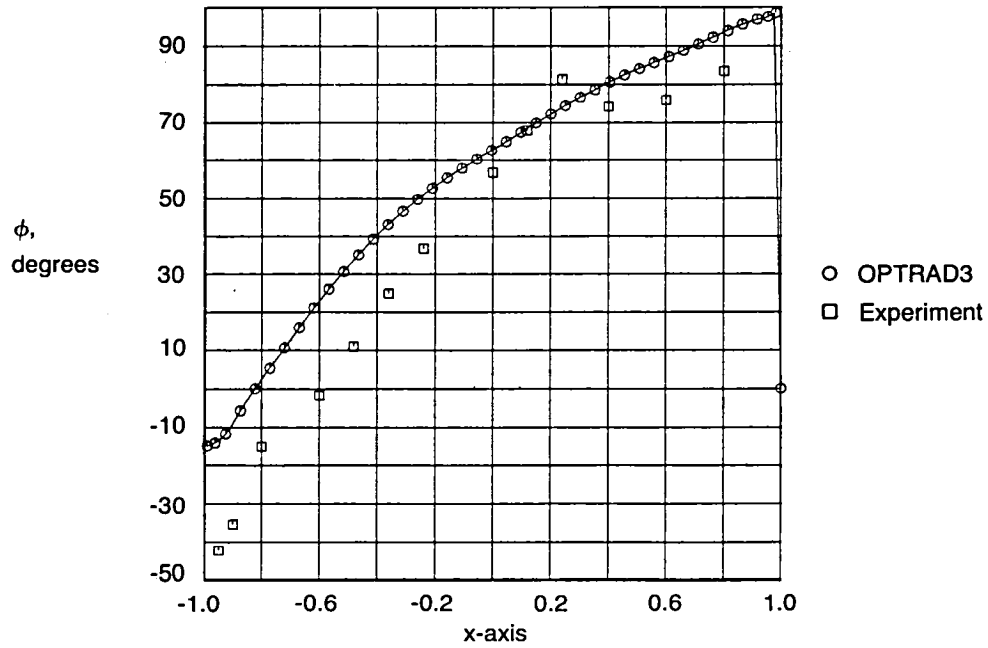


(a) 0.31 semispan

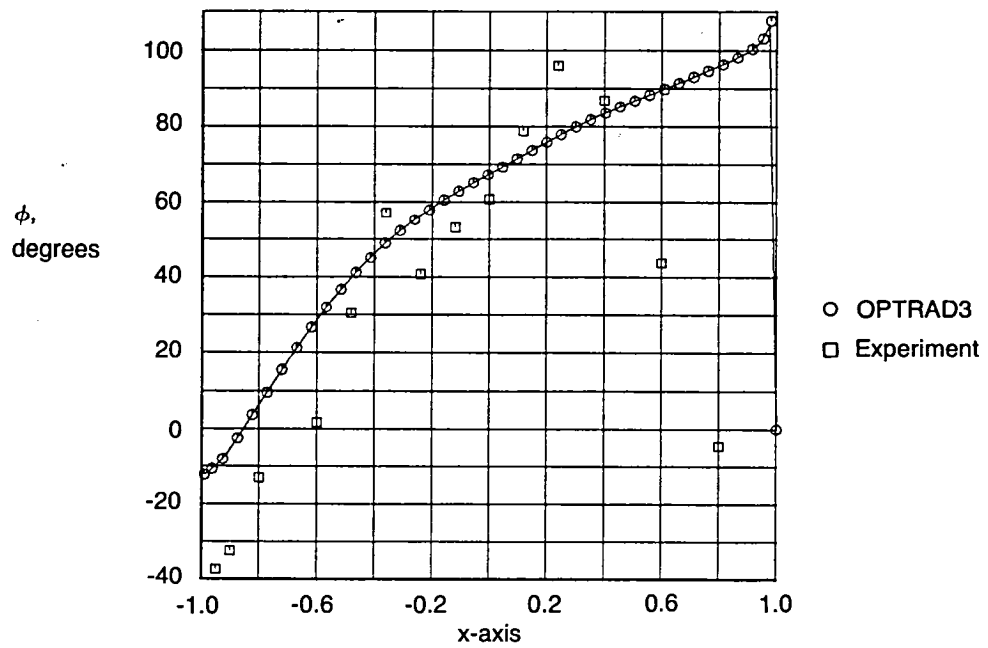


(b) 0.59 semispan

Figure 36.—Correlation of OPTRAD3 With Experimental Result, Pressure Phase-Angle Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, $M = 0.7$, $k = 0.356$

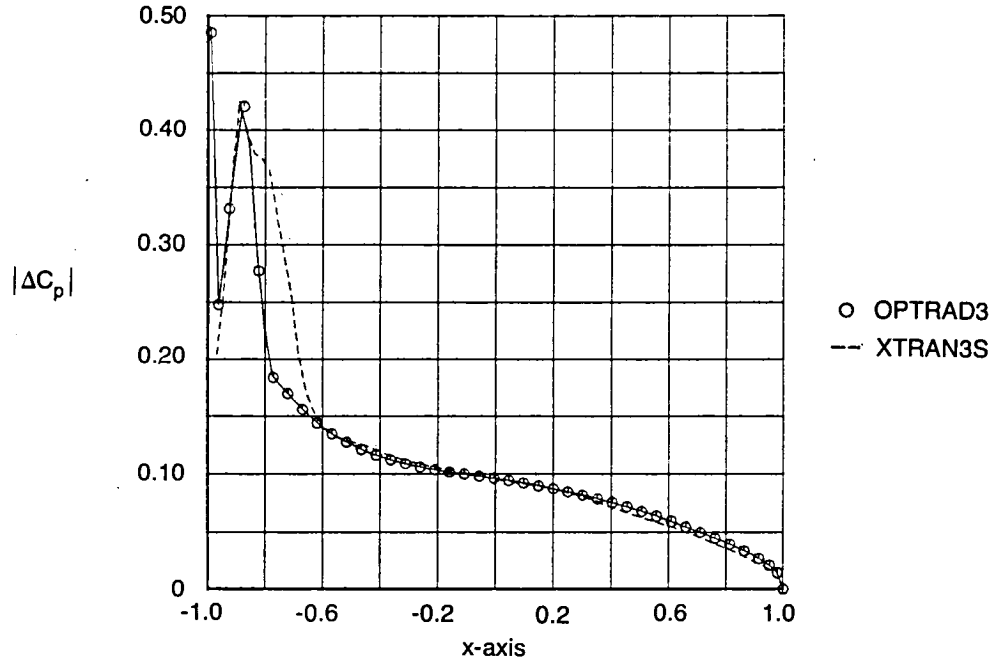


(c) 0.81 semispan

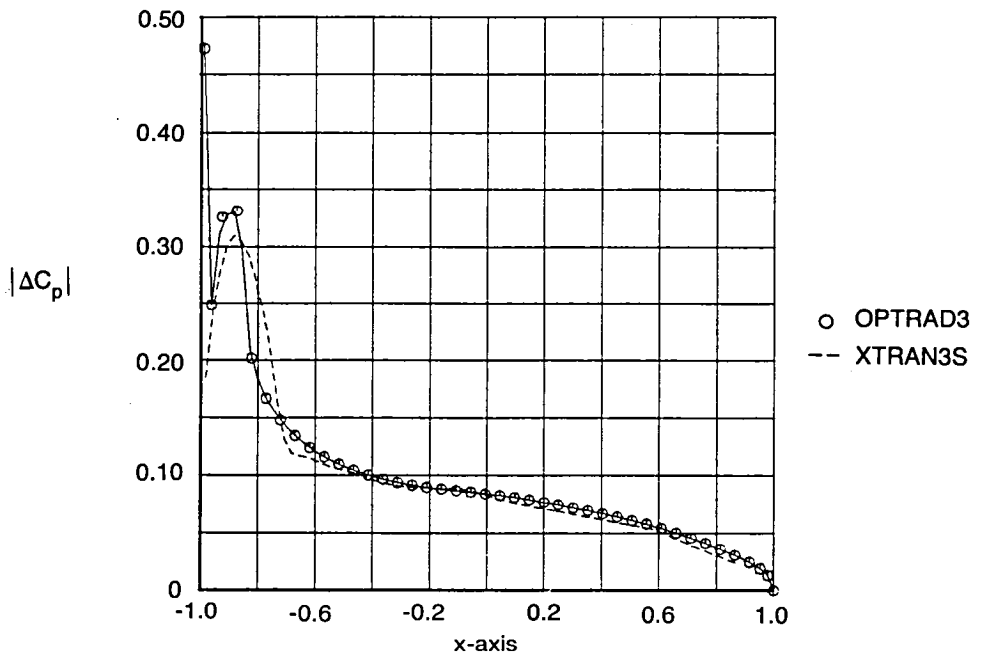


(d) 0.95 semispan

Figure 36.—Correlation of OPTRAD3 With Experimental Result, Pressure Phase-Angle Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.356 (Concluded)

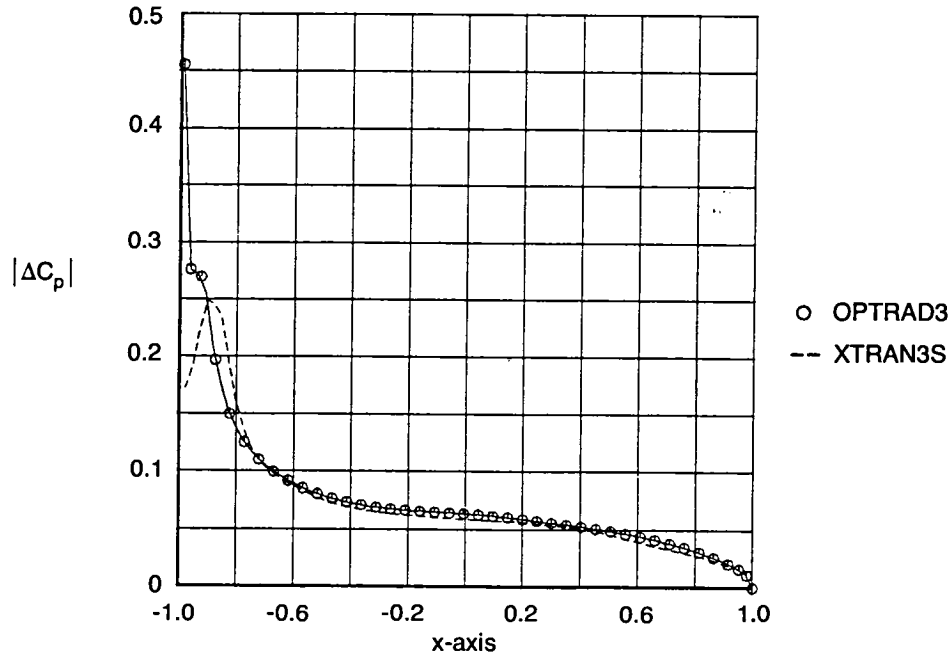


(a) 0.31 semispan

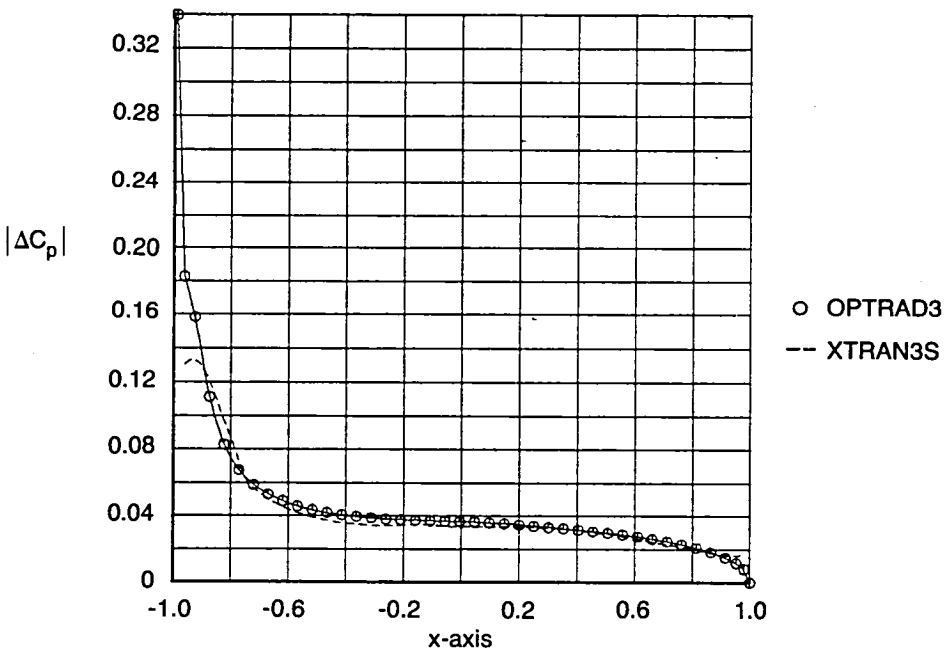


(b) 0.59 semispan

Figure 37.—Comparison of OPTRAD3 With XTRAN3S, Pressure Amplitude Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.356

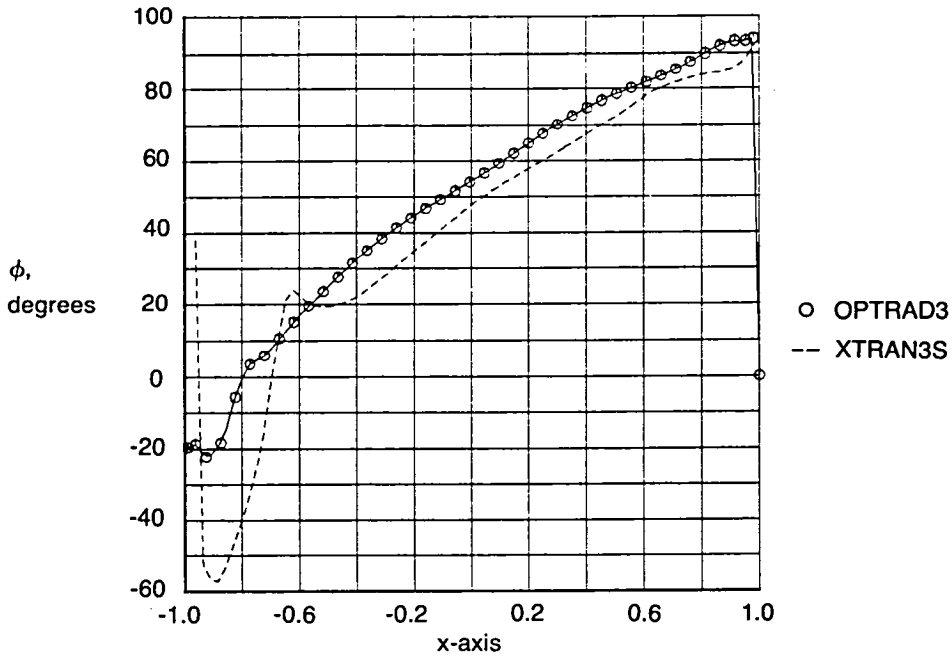


(c) 0.81 semispan

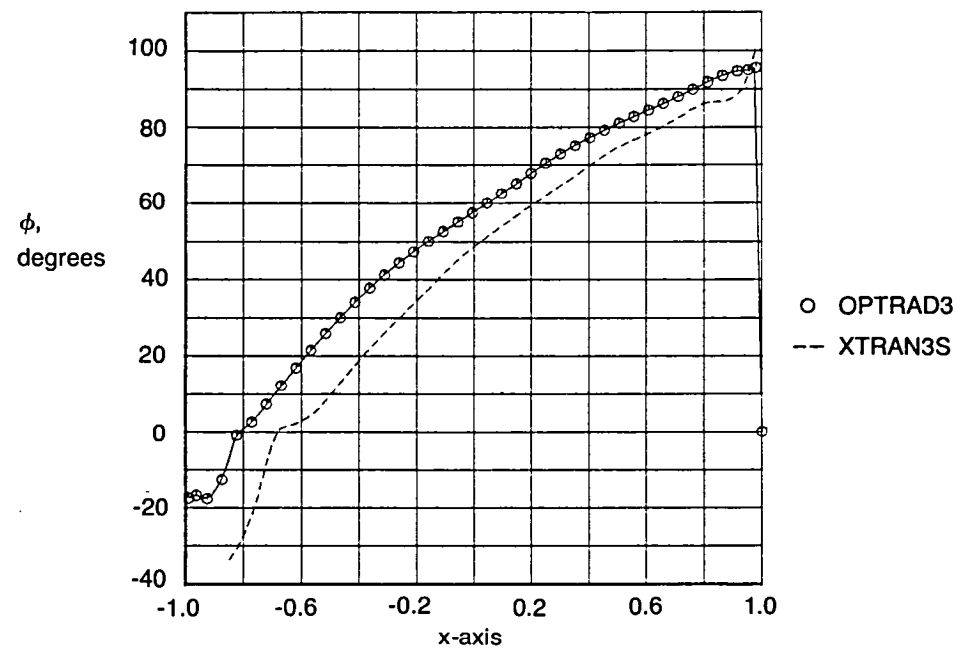


(d) 0.95 semispan

Figure 37.—Comparison of OPTRAD3 With XTRAN3S, Pressure Amplitude Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.356 (Concluded)

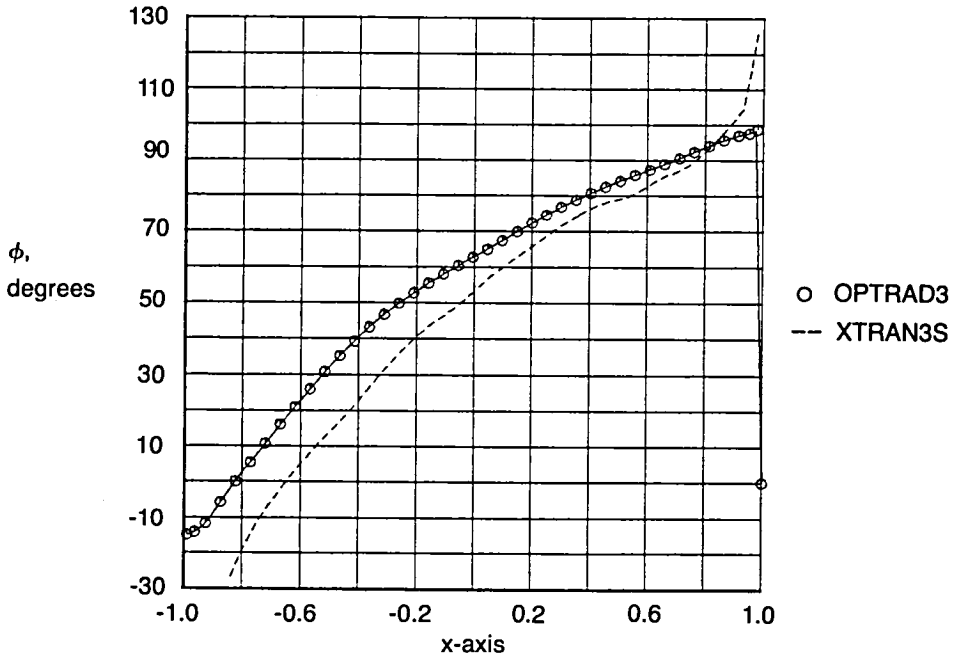


(a) 0.31 semispan

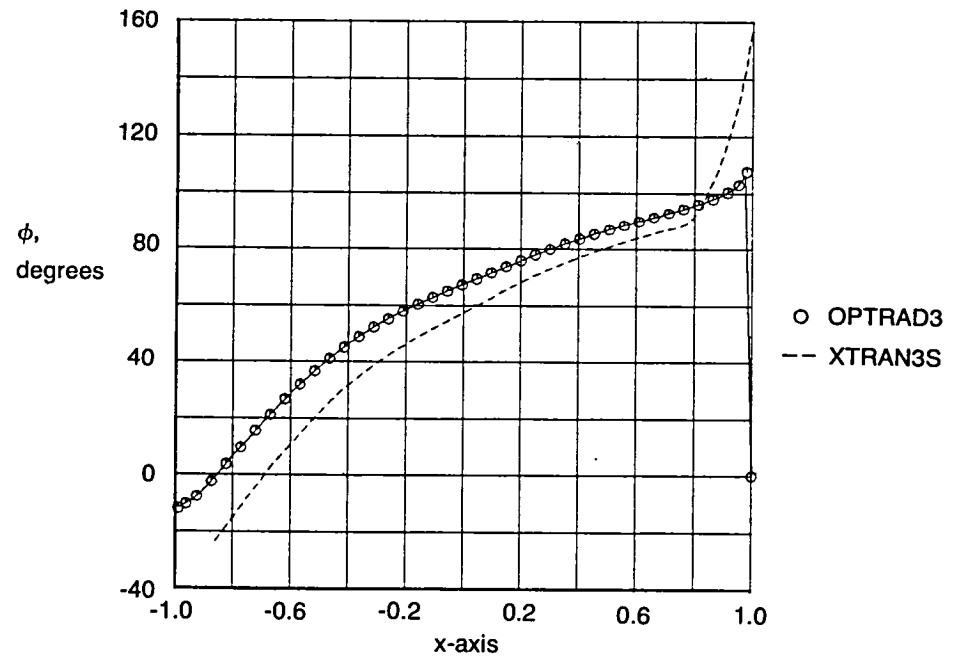


(b) 0.59 semispan

Figure 38.—Comparison of OPTRAD3 With XTRAN3S, Pressure Phase-Angle Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.356



(c) 0.81 semispan



(d) 0.95 semispan

Figure 38.—Comparison of OPTRAD3 With XTRAN3S, Pressure Phase-Angle Distributions, Aspect Ratio 4 Rectangular Wing With Supercritical Airfoil, M = 0.7, k = 0.356 (Concluded)

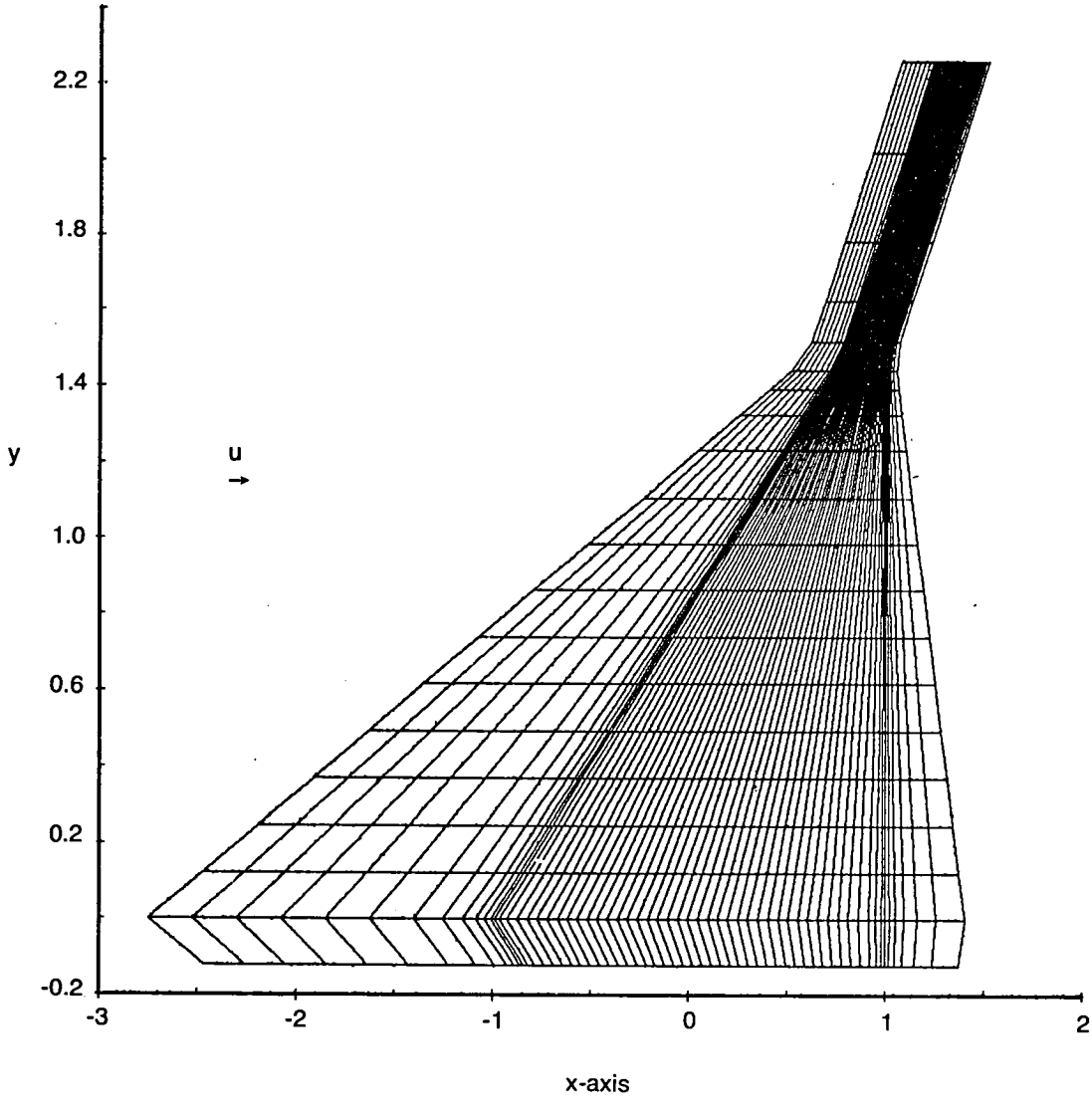


Figure 39.—Example of Swept Wing Coordinate System in the Physical Plane for a Clipped Delta Planform

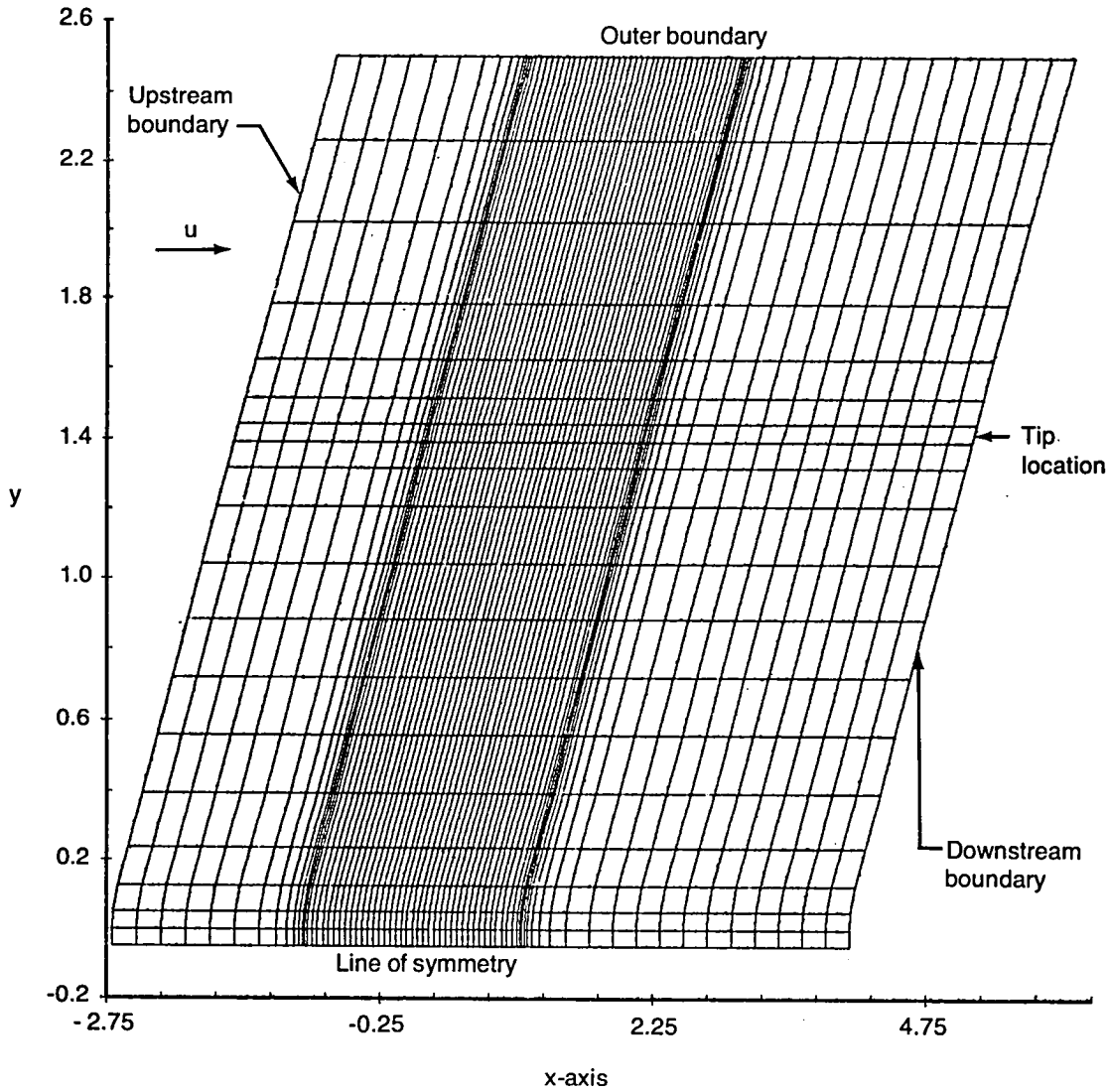


Figure 40.—Example of Swept Wing Coordinate System in the Physical Plane for an Untapered Swept Wing With Modified Root Geometry

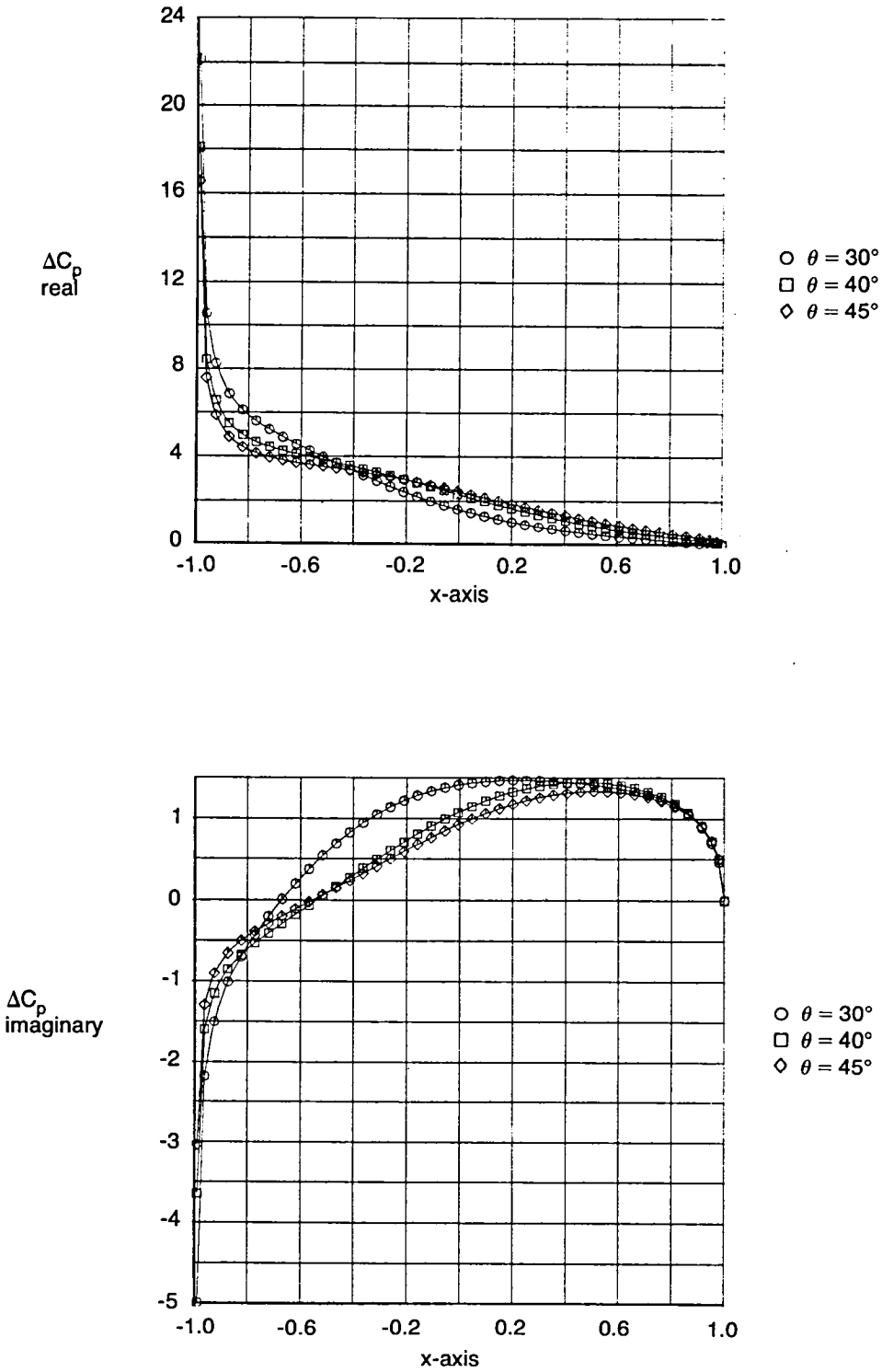


Figure 41.—OPTRAD3 Pressure Distributions for an Untapered Wing With Three Angles of Sweep, M = 0.9, k = 0.13, Root Chord

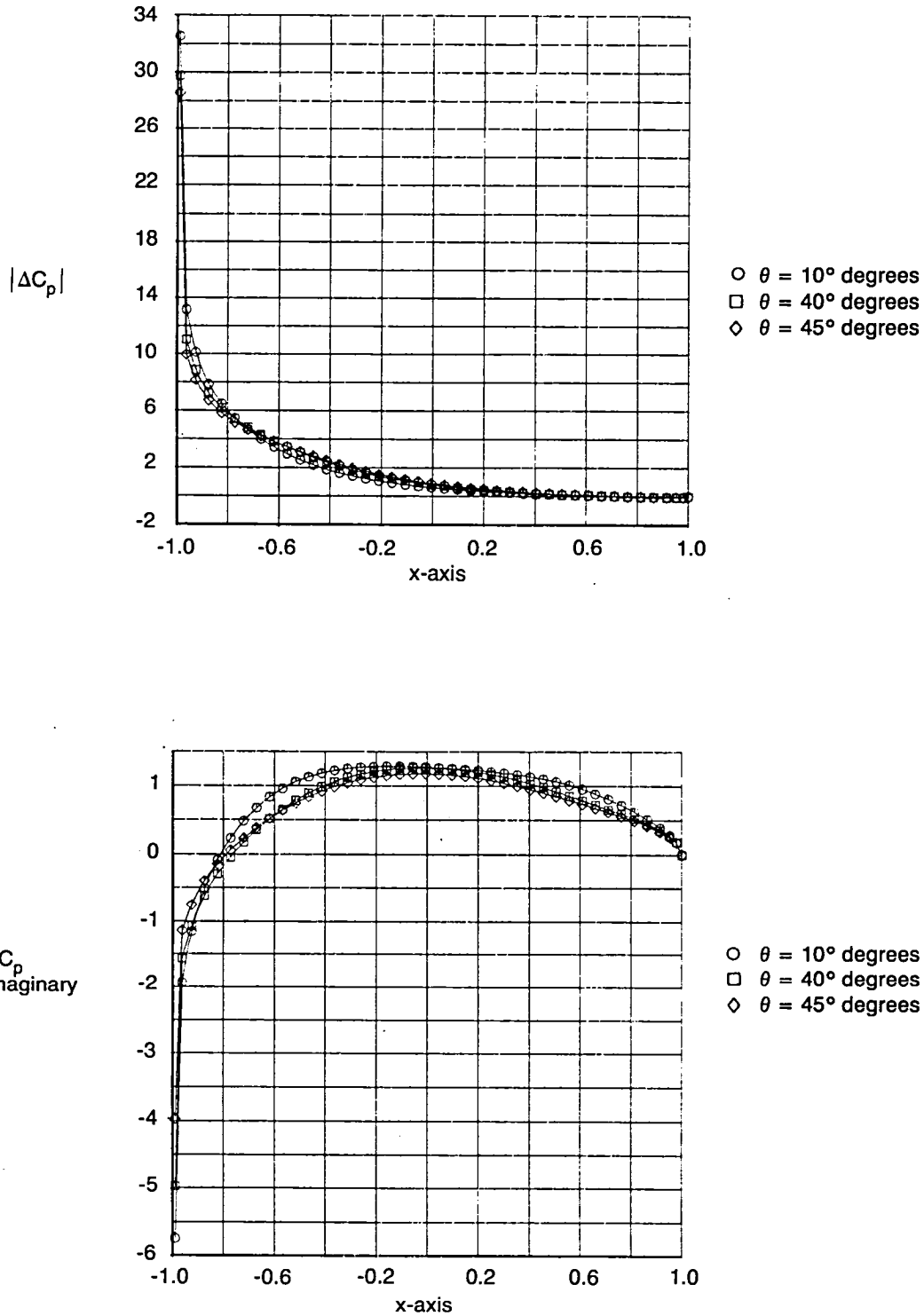


Figure 42.—OPTRAD3 Pressure Distributions for an Untapered Wing With Three Angles of Sweep, $M = 0.9$, $k = 0.13$, $\bar{\eta} = 0.51$

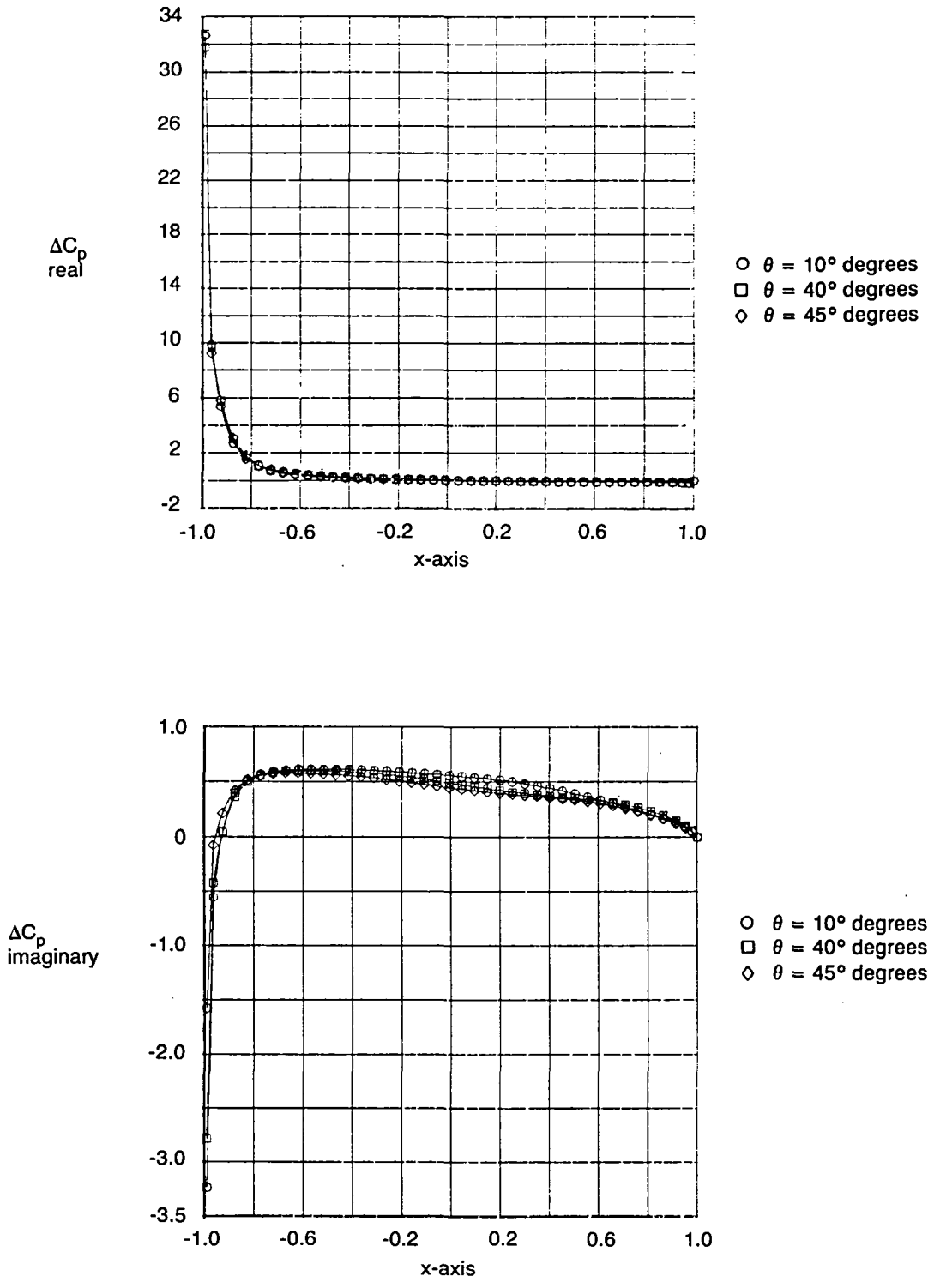


Figure 43.—OPTRAD3 Pressure Distributions for an Untapered Wing With Three Angles of Sweep, $M = 0.9$, $k = 0.13$, $\bar{\eta} = 0.93$

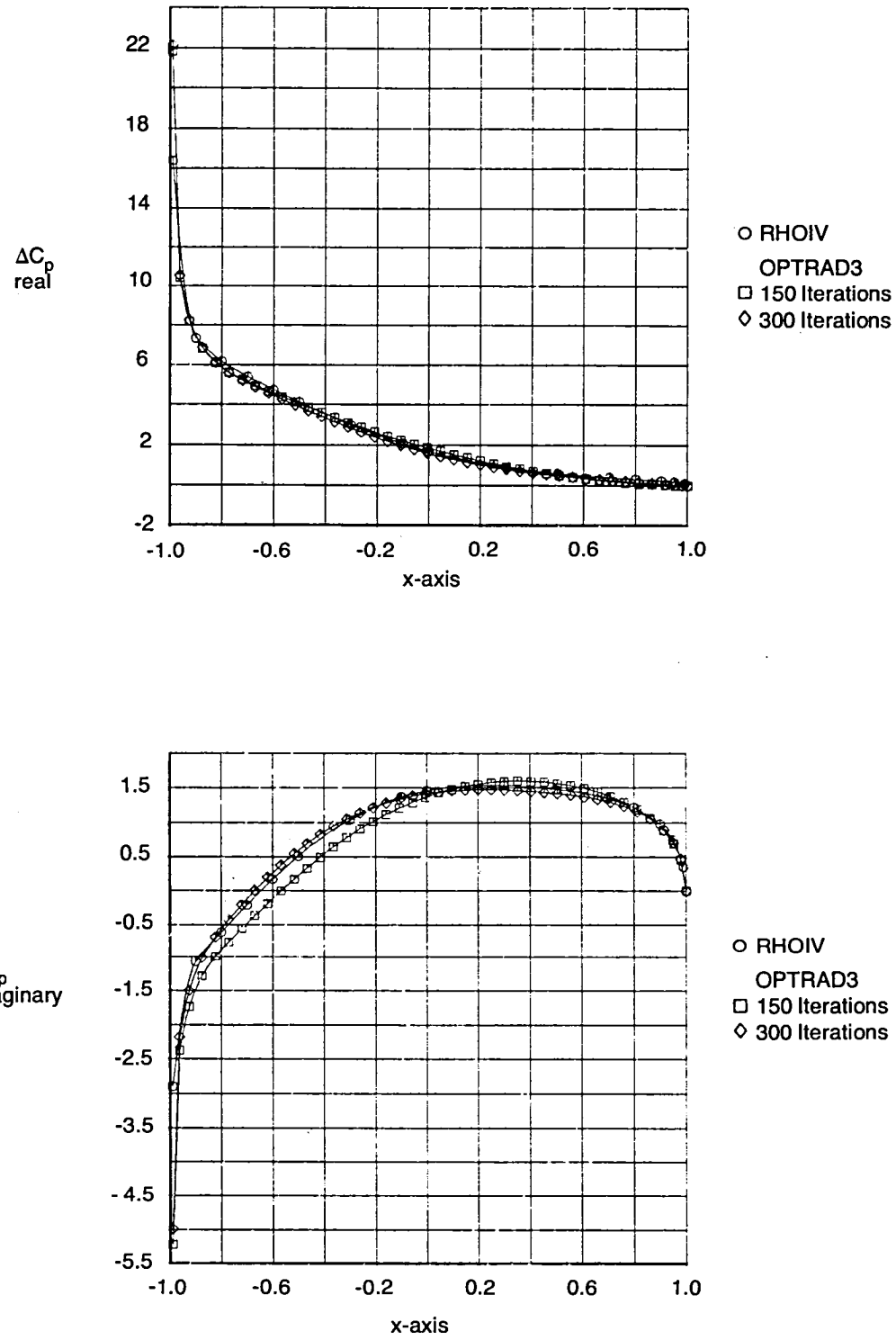


Figure 44.—Comparison of OPTRAD3 With RHOIV, Pressure Distributions for a 30-deg Swept, Untapered Wing, M = 0.9, k = 0.13, Root Chord

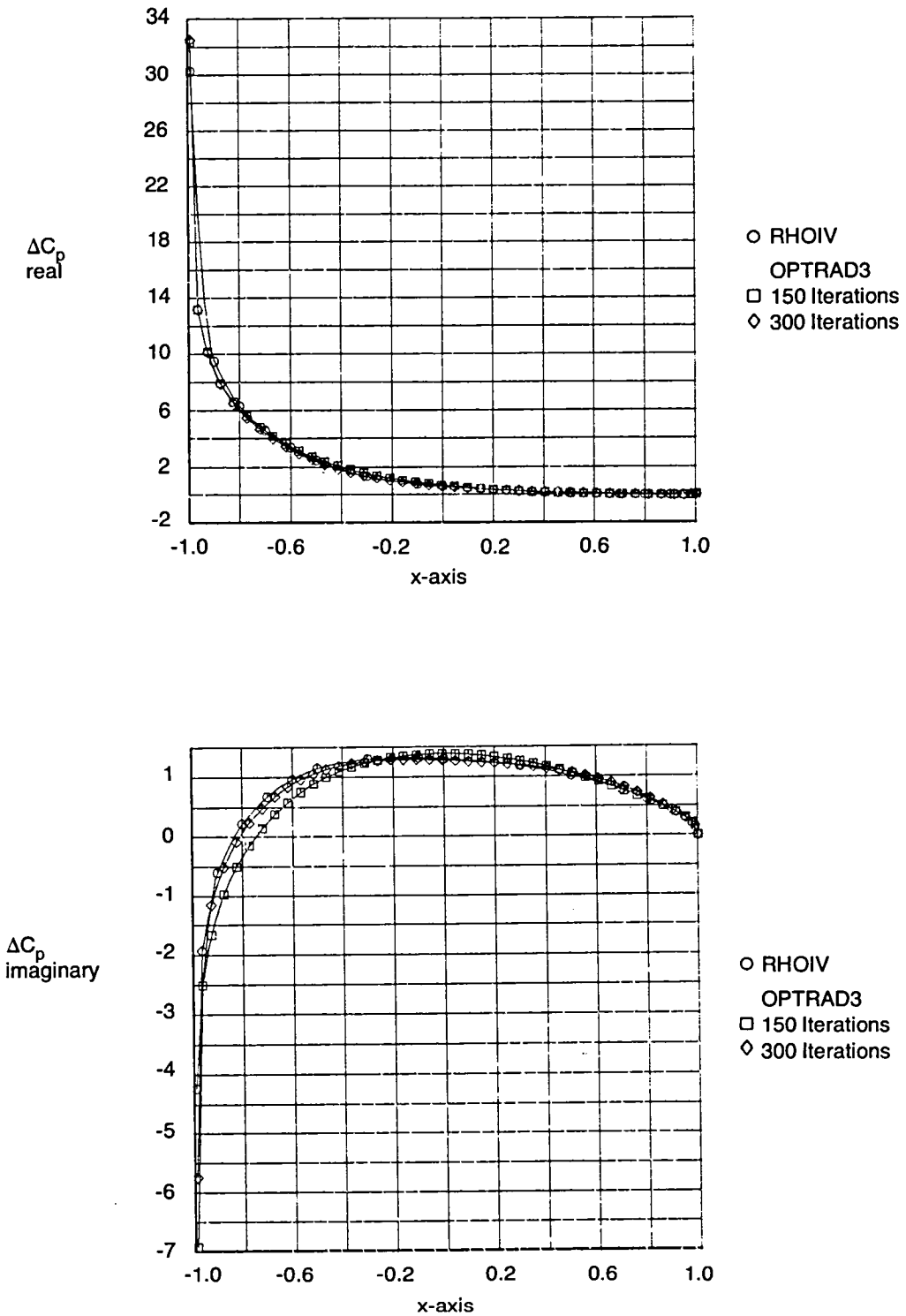


Figure 45.—Comparison of OPTRAD3 With RHOIV, Pressure Distributions for a 30-deg Swept, Untapered Wing, $M = 0.9$, $k = 0.13$, $\bar{\eta} = 0.51$

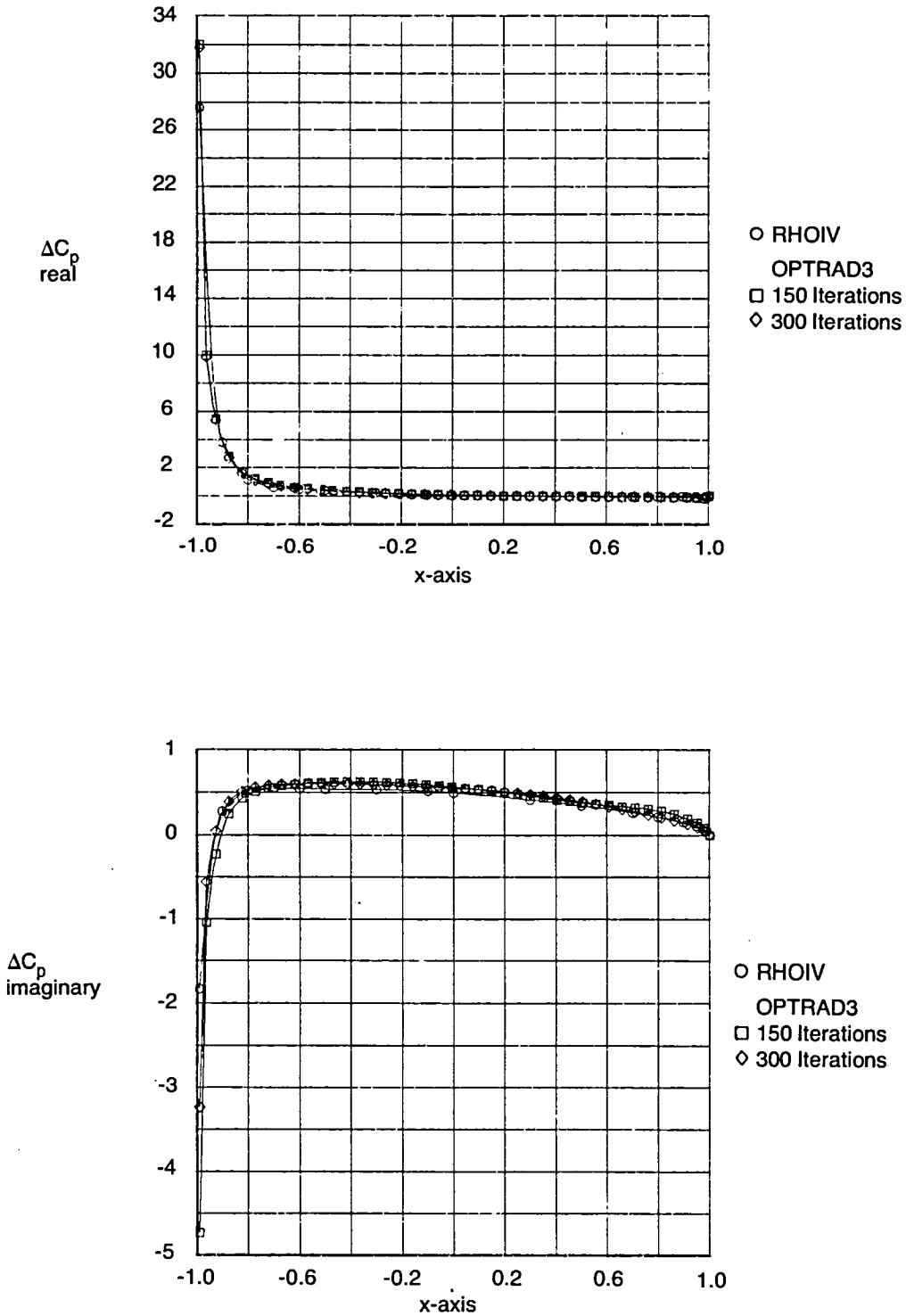


Figure 46.—Comparison of OPTRAD3 With RHOIV, Pressure Distributions for a 30-deg Swept, Untapered Wing, $M = 0.9$, $k = 0.13$, $\bar{\eta} = 0.93$.

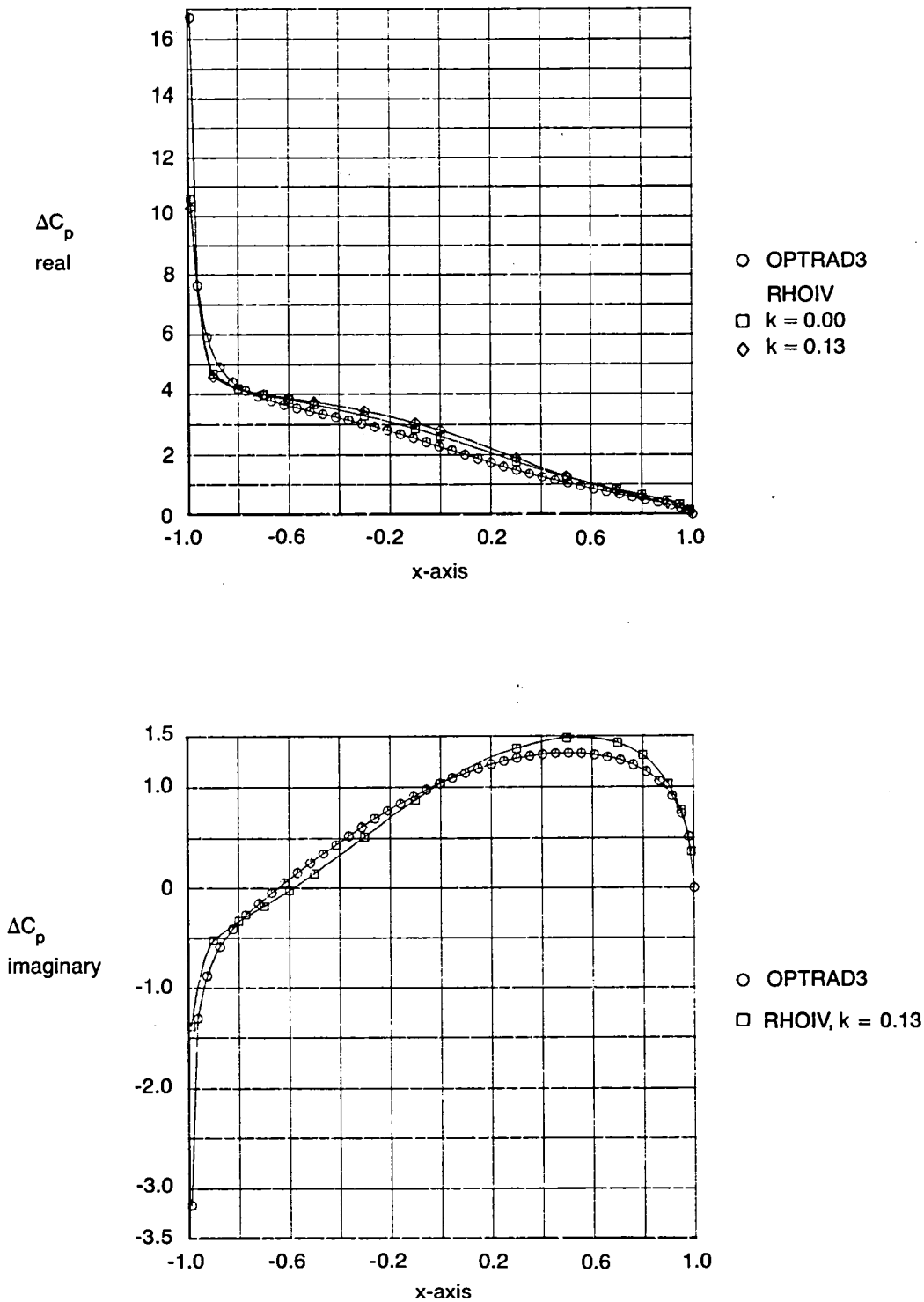


Figure 47.—Comparison of OPTRAD3 With RHOIV, Pressure Distributions for a 45-deg Swept, Untapered Wing, M = 0.9, k = 0.13, Root Chord

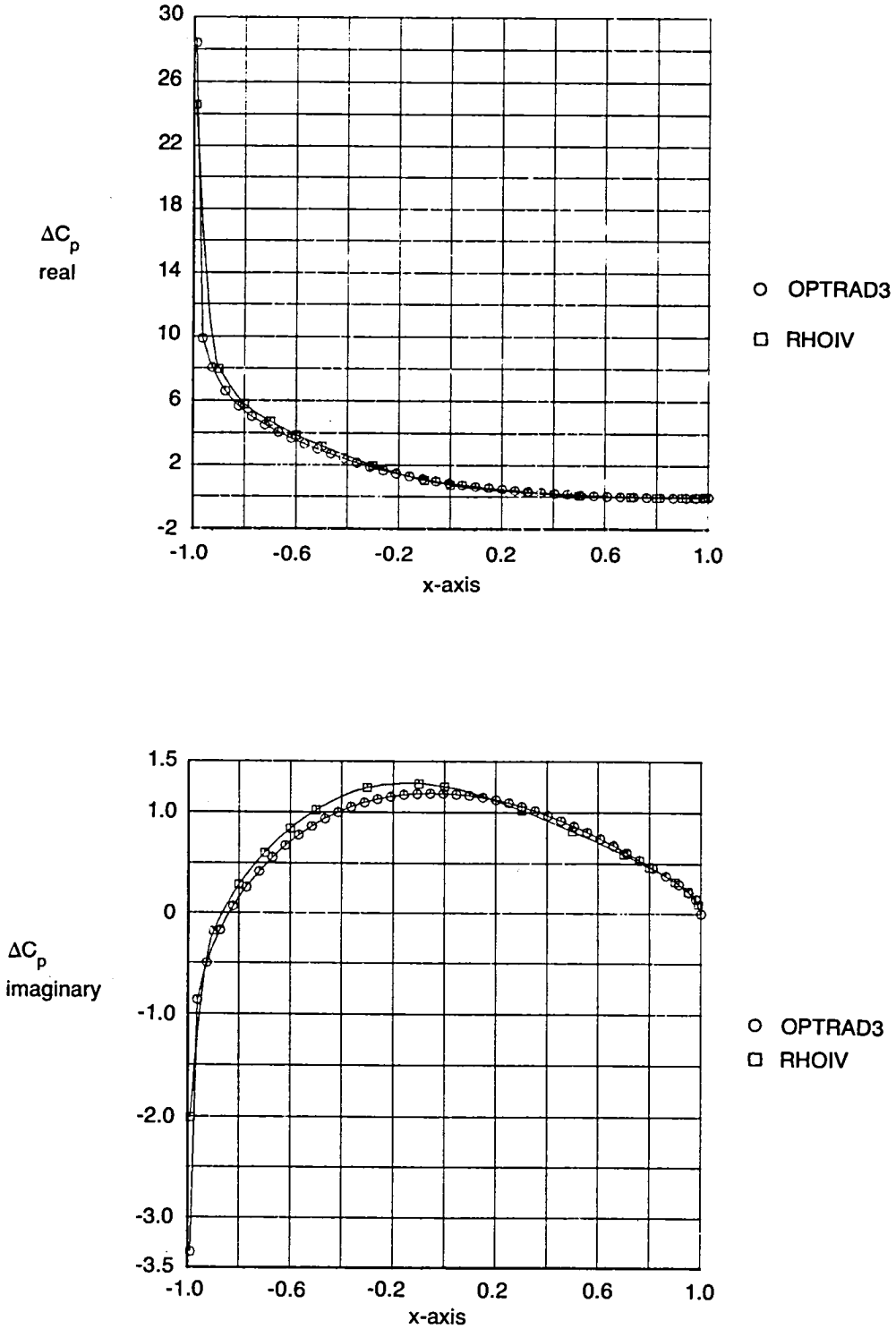
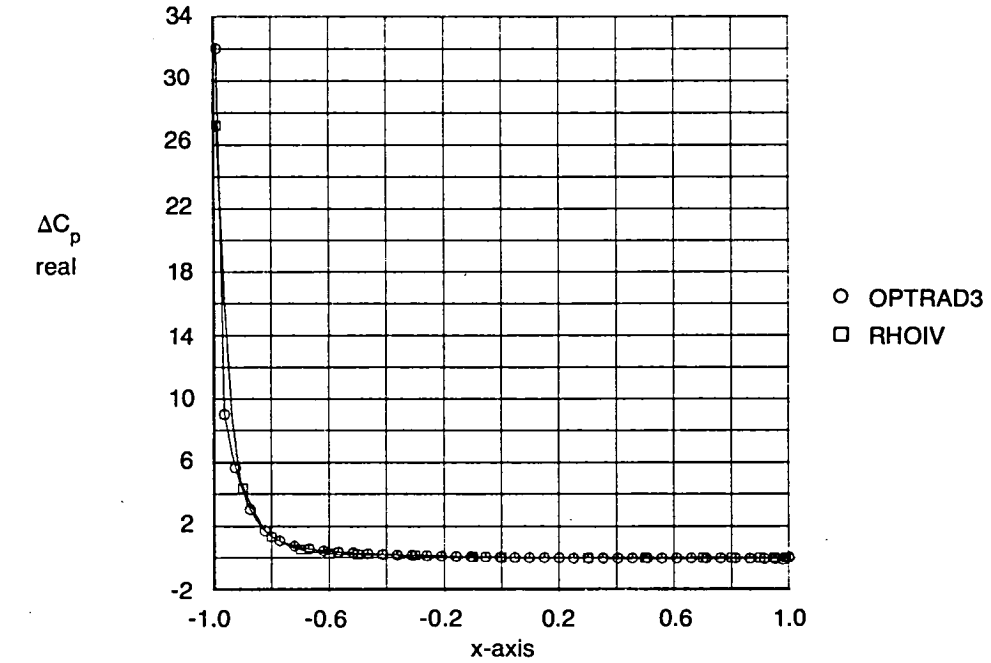


Figure 48.—Comparison of OPTRAD3 With RHOIV, Pressure Distributions for a 45-deg Swept, Untapered Wing, $M = 0.9$, $k = 0.13$, $\bar{\eta} = 0.51$



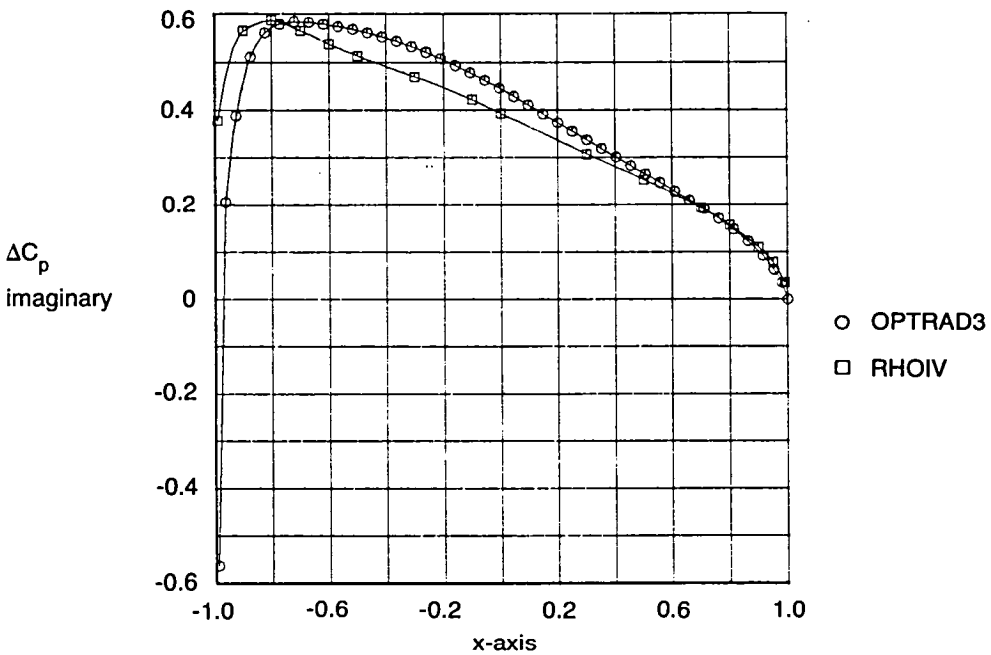


Figure 49.—Comparison of OPTRAD3 With RHOIV, Pressure Distributions for a 45-deg Swept, Untapered Wing, $M = 0.9$, $k = 0.13$, $\bar{\eta} = 0.93$

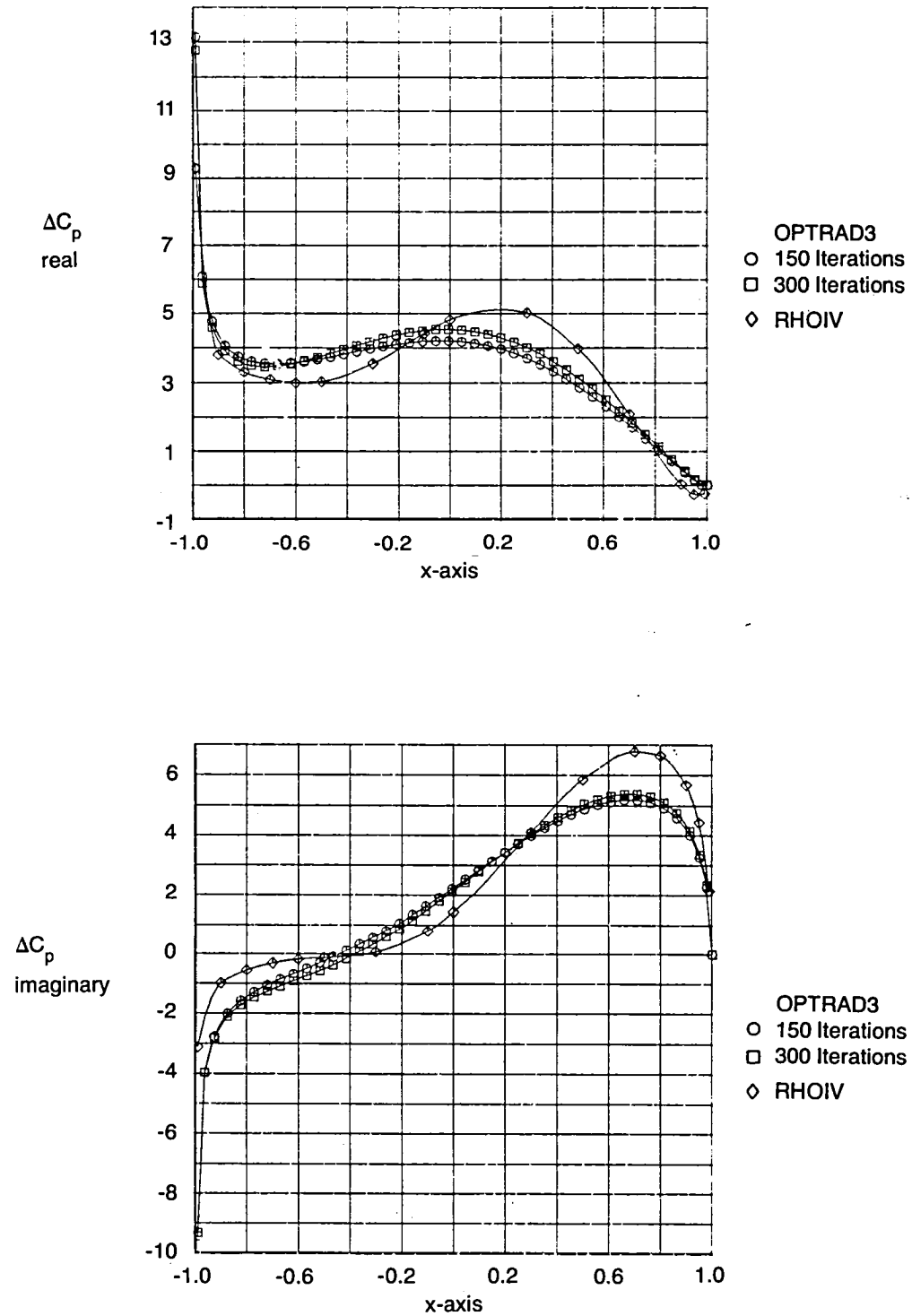


Figure 50.—Comparison of OPTRAD3 With RHOIV, Pressure Distributions for a 45-deg Swept, Untapered Wing, M = 0.9, k = 0.5, Root Chord

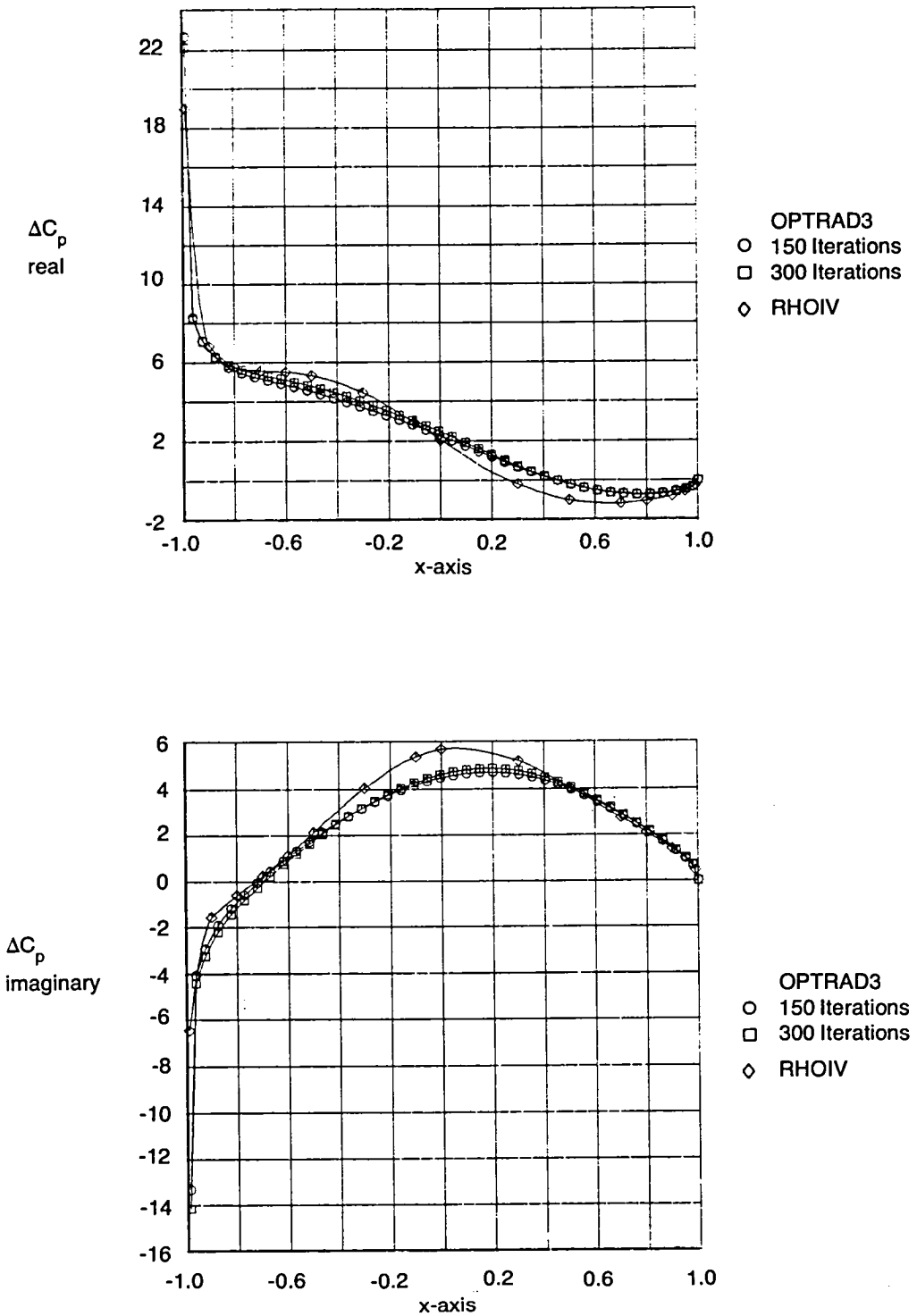


Figure 51.—Comparison of OPTRAD3 With RHOIV, Pressure Distributions for a 45-deg Swept, Untapered Wing, $M = 0.9$, $k = 0.5$, $\bar{\eta} = 0.51$

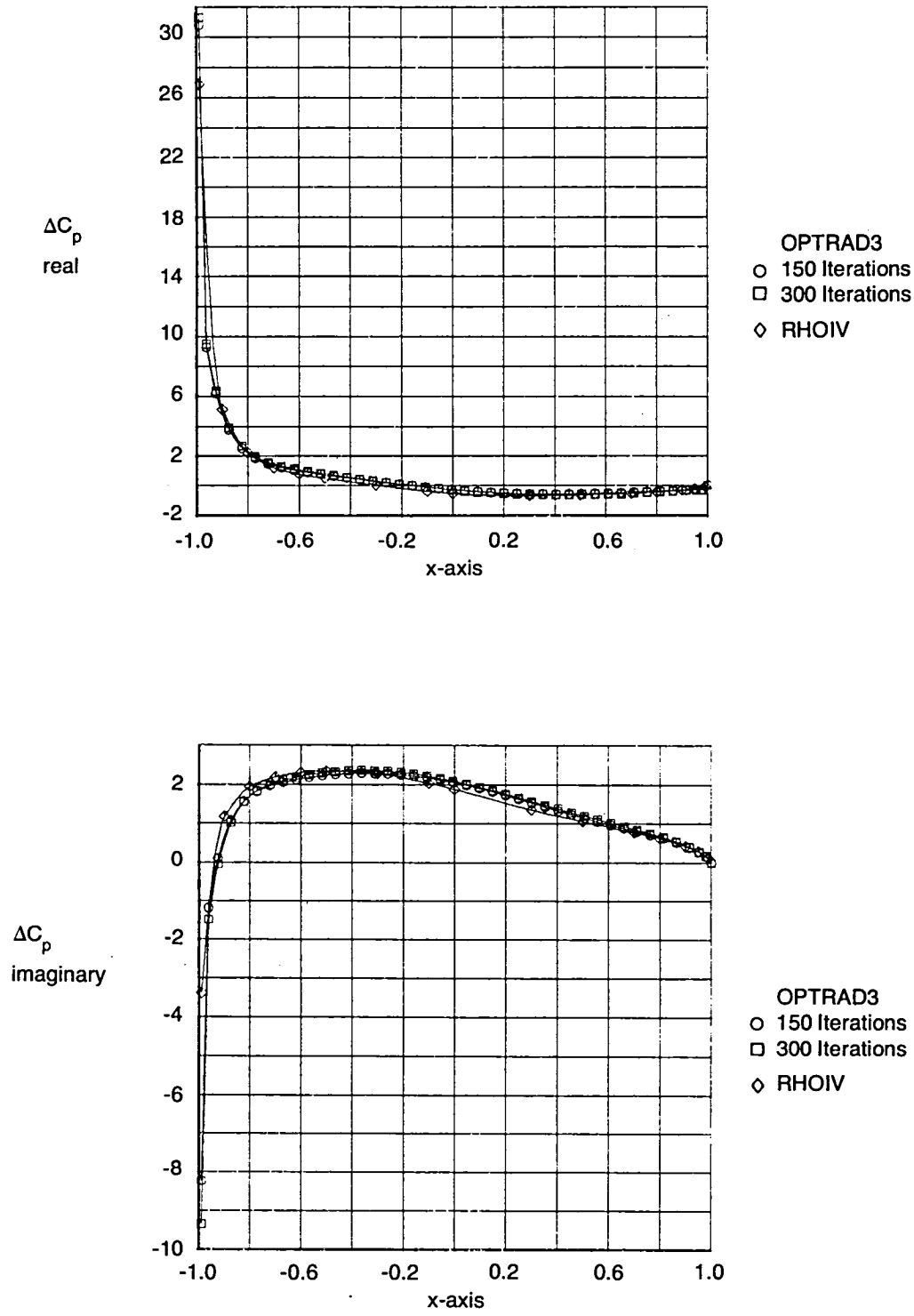


Figure 52.—Comparison of OPTRAD3 With RHOIV, Pressure Distributions of a 45-deg Swept, Untapered Wing, $M = 0.9$, $k = 0.5$, $\bar{\eta} = 0.93$

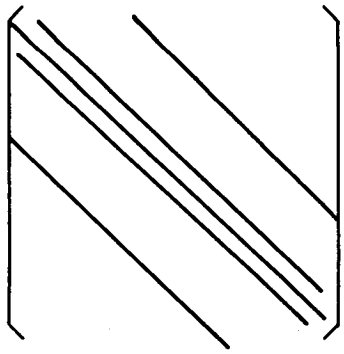


Figure 53.—Sparsity Structure of A

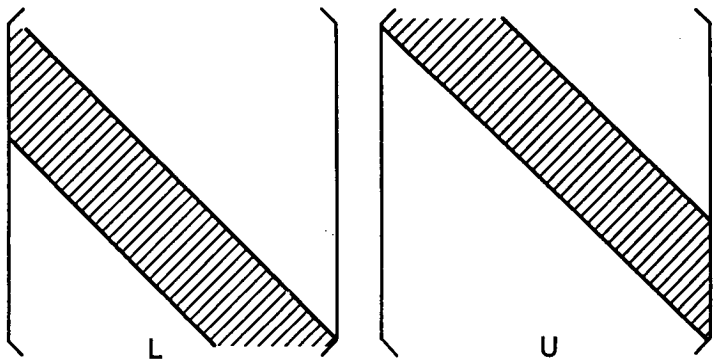


Figure 54.—Sparsity Structure of LU = A

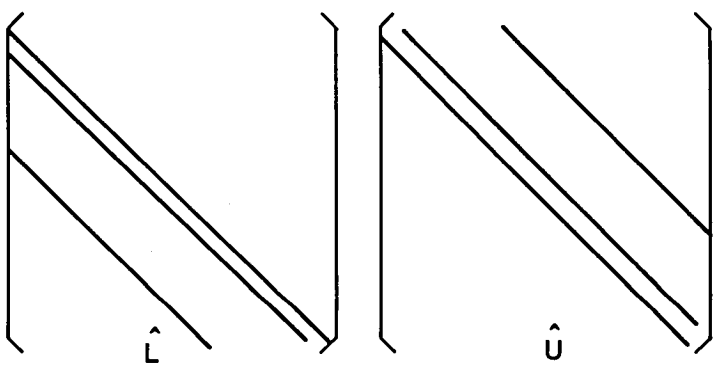


Figure 55.—Sparsity Structure of Incomplete LU Factorization

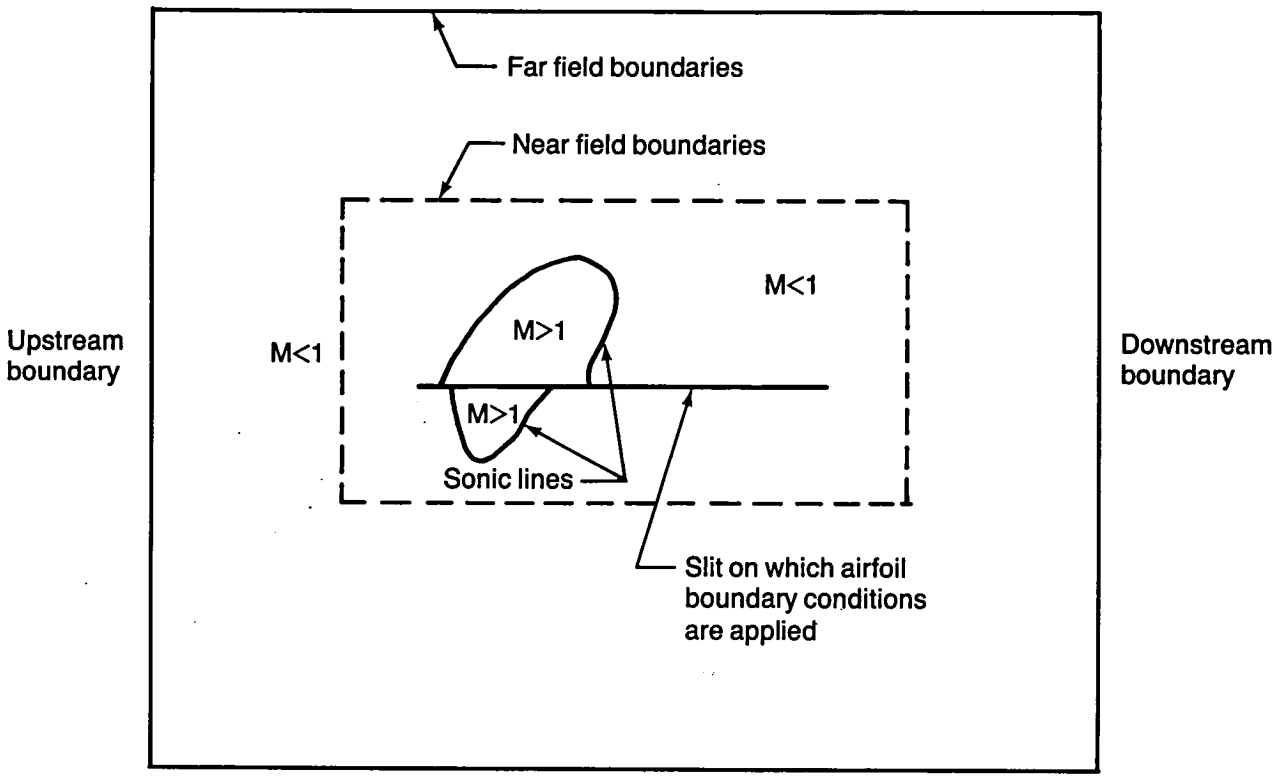


Figure 56.—Near and Far Field Boundaries for the Sparse Capacitance Matrix Method for Two-Dimensional Flow

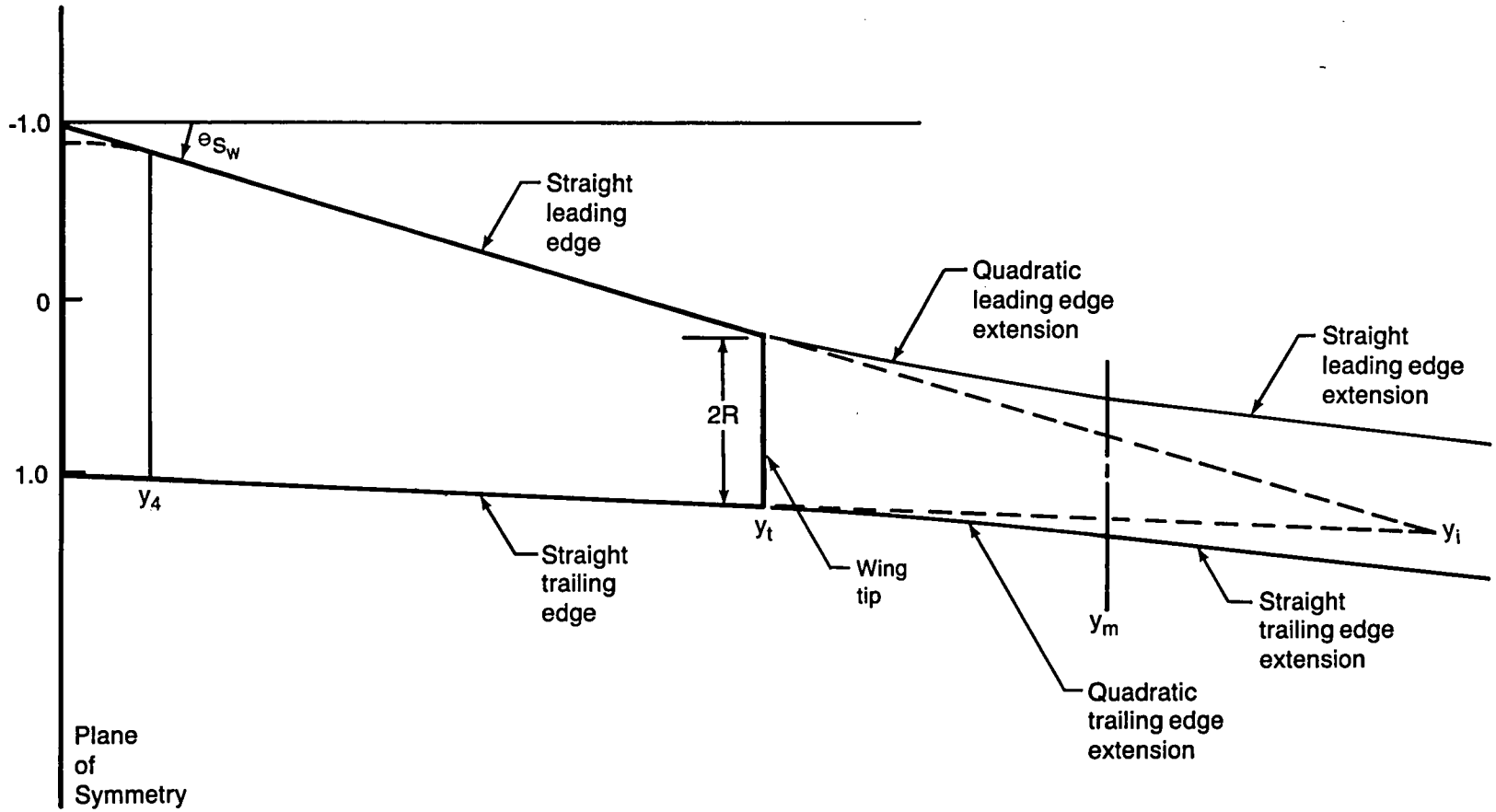


Figure 57.—Illustration of the Extension of Wing Leading and Trailing Edge Coordinate Lines for Deriving the Swept Wing Transformation

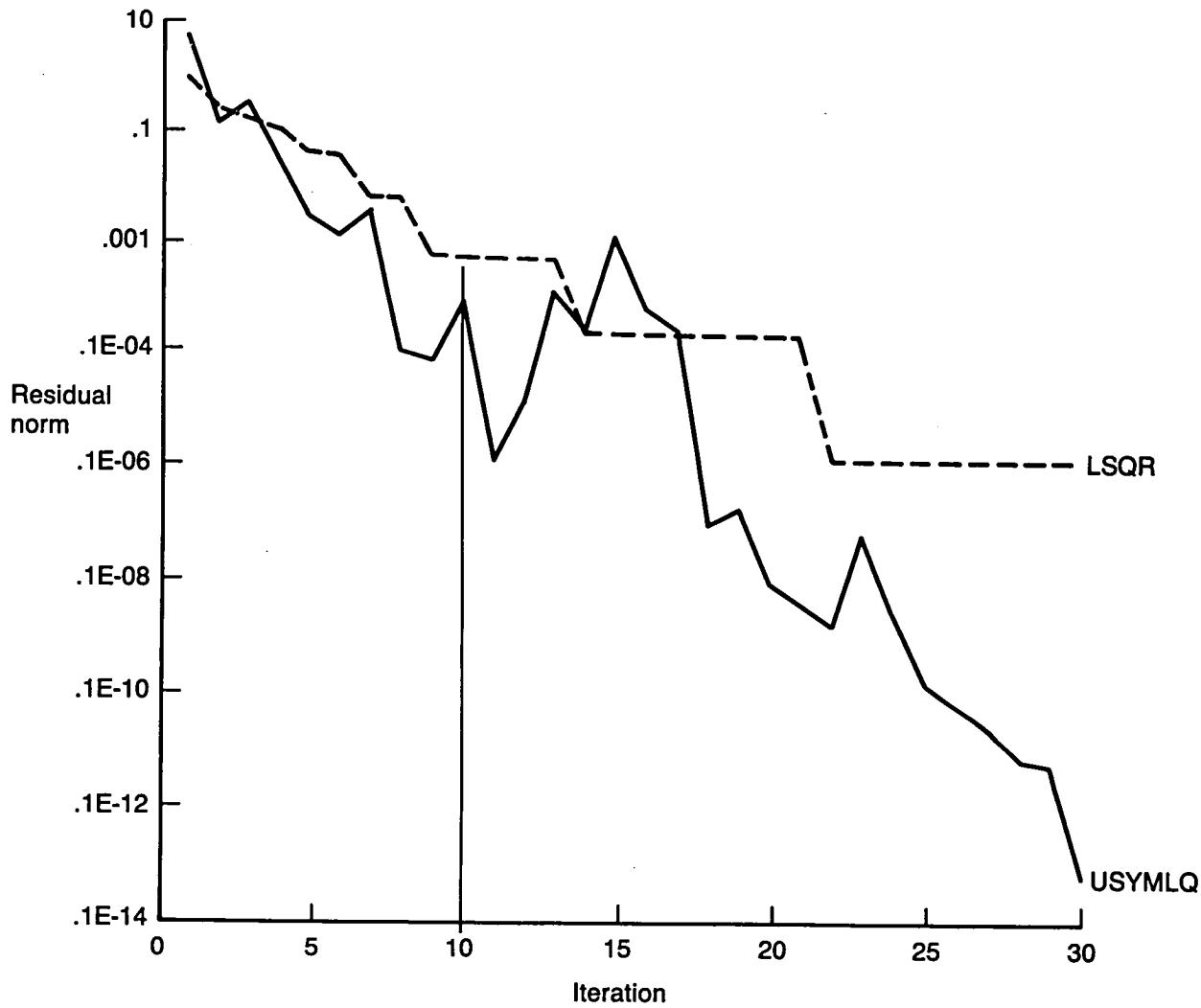


Figure 58.—Comparisons of Convergence Performance of USYMLQ and LSQR, N = 10, K = 10^8, 5 = 10

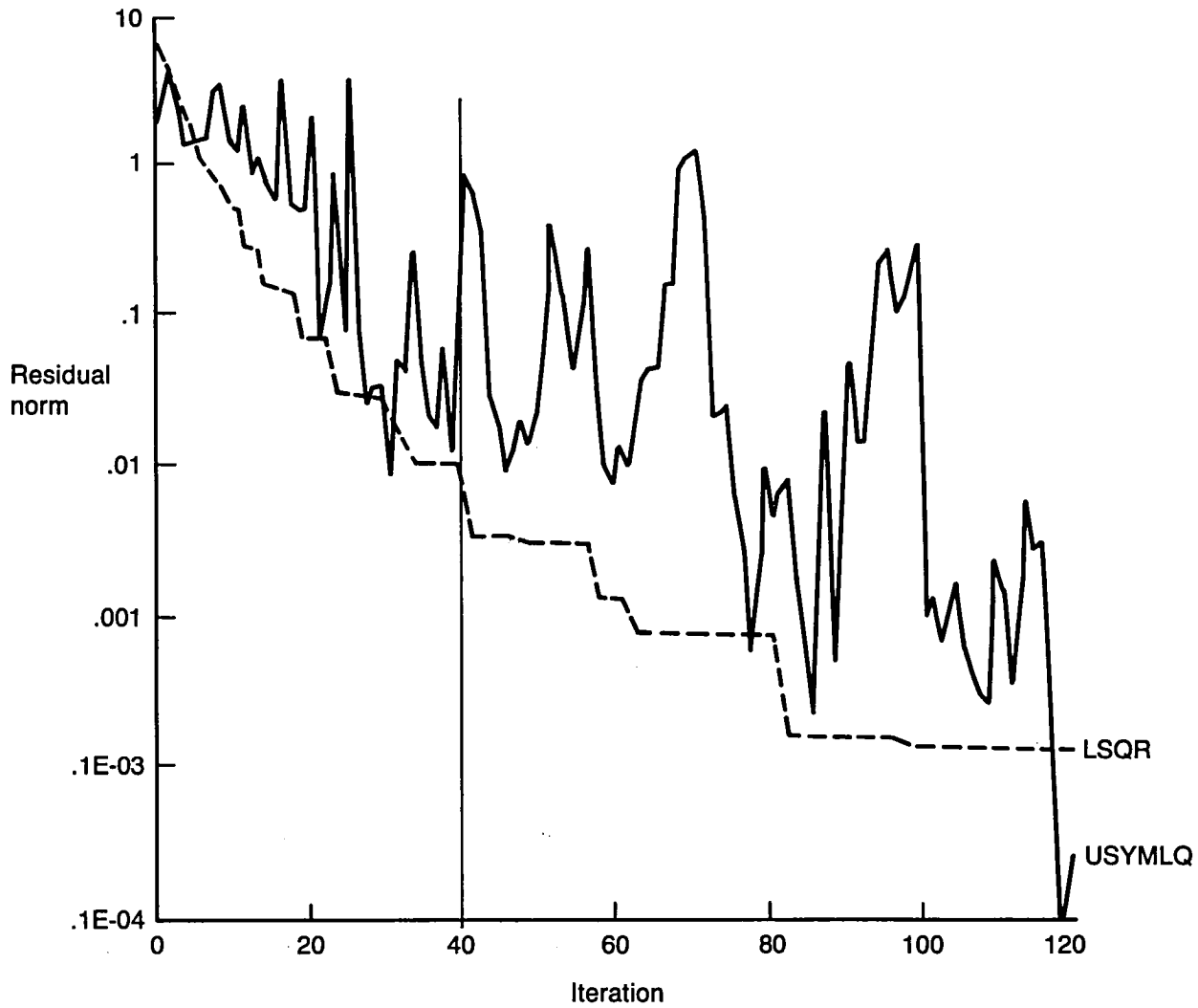


Figure 59.—Comparisons of Convergence Performance of USYMLQ and LSQR, N = 40, $K = 9.7 \times 10^9$, S = 20

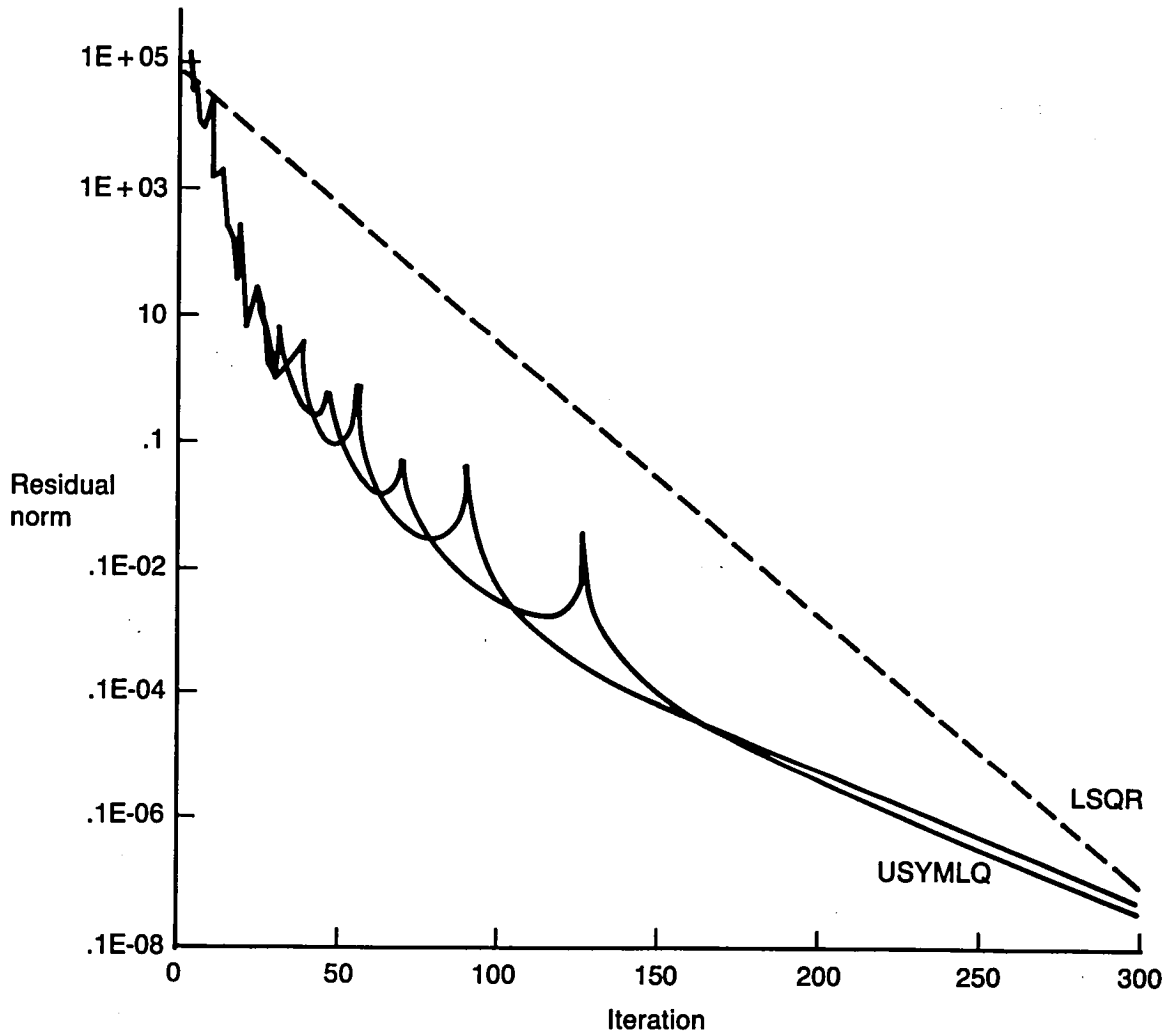


Figure 60.—Comparisons of Convergence Performance of USYMLQ and LSQR, N = 2000, K = 20, S = 2000.

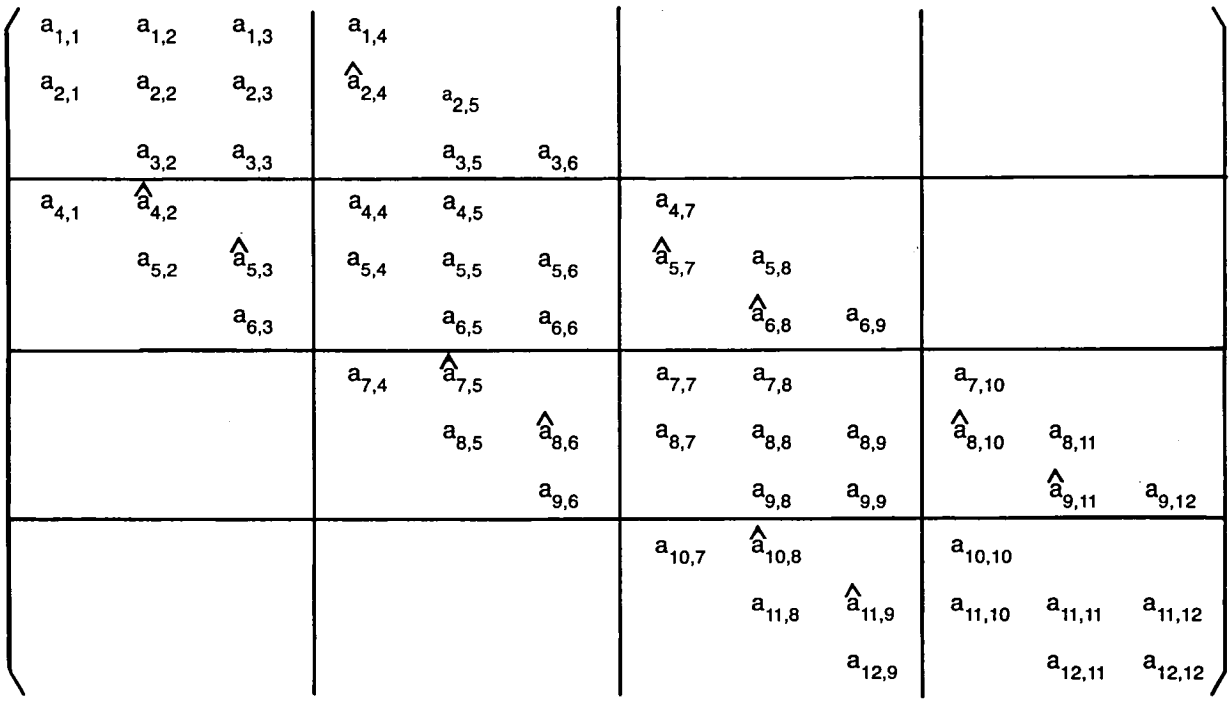


Figure 61.—Incomplete LU Factorization

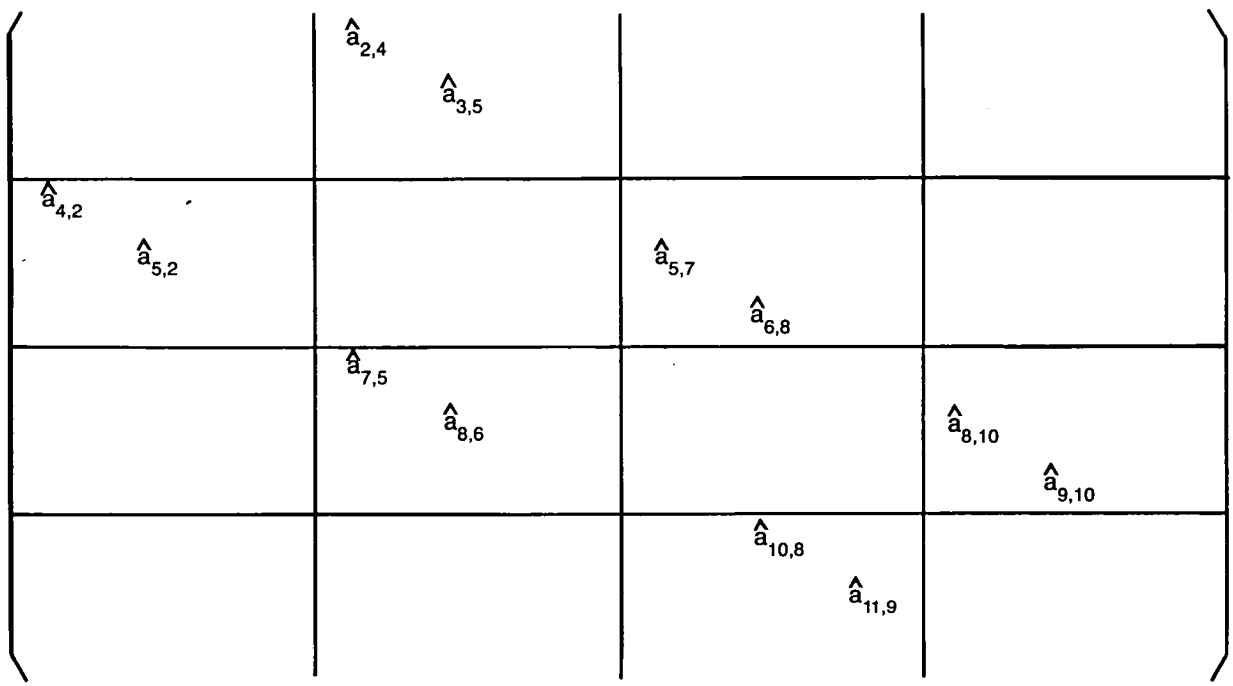


Figure 62.—Remainder of Incomplete LU Factorization

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR 172376		

4. Title and Subtitle	5. Report Date
Development and Applications of Algorithms for Calculating the Transonic Flow About Harmonically Oscillating Wings	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
F. E. Ehlers, W. H. Weatherill and E. L. Yip	
9. Performing Organization Name and Address	10. Work Unit No.
Boeing Commercial Airplane Company P. O. Box 3707 Seattle, WA 98112	11. Contract or Grant No. NAS1 - 16297
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Langley Research Center National Aeronautics and Space Administration Washington D. C. 20546	Contractor Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor: Robert M. Bennett

16. Abstract

A finite difference method for solving the unsteady transonic flow about harmonically oscillating wings is investigated. The procedure is based on separating the velocity potential into steady and unsteady parts and linearizing the resulting unsteady differential equation for small disturbances. The differential equation for the unsteady velocity potential is linear with spatially varying coefficients and with the time variable eliminated by assuming harmonic motion.

An alternating direction implicit procedure is investigated, and a pilot program is developed for both two-and three-dimensional wings. This program provides a relatively efficient relaxation solution without previously encountered solution instability problems.

Pressure distributions for two rectangular wings are calculated and the results correlated with experimental data.

Conjugate gradient techniques are developed for the asymmetric, indefinite problem. The conjugate gradient procedure is evaluated for applications to the unsteady transonic problem. Several preconditioning methods are investigated.

Finally, different equations for the alternating direction procedure are derived using a coordinate transformation for swept and tapered wing planforms. Pressure distributions for swept, untapered wings of vanishing thickness are correlated with linear results for sweep angles up to 45 degrees.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Unsteady Flow Transonic Flow Oscillating Airfloils	Unclassified - Unlimited Subject Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	150	

For sale by the National Technical Information Service, Springfield, Virginia 22151